UNIVERSITY OF OSLO
Department of Informatics

Internationalisation in Health Information System Project

Research in Vietnamese Language

Master thesis

Teresa Hang Nguyen

19th December 2005

Abstract

This thesis is based on an action research project where I have participated in the development of internationalisation support to a computerised health information system. The project is a Vietnamese-Norwegian collaboration targeted at implementing and adapting the District Health Information Software (DHIS) in Vietnam, part of the larger Health Information Systems Programme (HISP). HISP is a global research initiative that started in South Africa in 1994 and has since been implemented in many other developing countries, now recently in Vietnam.

Following an action research approach, the author has actively participated in the information system development process to enable multi-language support in the software and to explore the importance of internationalisation in the context of information systems in developing countries. The author seeks to identify challenges related to developing internationalised software in this context, particularly in the Vietnamese language. The adaptation of DHIS to the Vietnamese context has been challenging, especially enabling support for the Vietnamese language. Both political and technical issues have played a major role in this process.

The author argues that internationalisation of global software is of importance in order to improve the situation of implementing IT-based systems in developing countries. Based on the research on the Vietnamese adaptation process it is clear that internationalisation of software represents many challenges; appropriate software design, limitations related to the technical platform, and limited support for certain languages.

Acknowledgement

This thesis makes up the main part of my studies towards a Master degree at the Department of Informatics, University of Oslo. It is based on an action research project where I have participated in development of health information systems. My initial thought when discovering the possibility to do a master thesis in development of district health information system (DHIS) software for developing country was something I wanted to do. As it would be a new and exciting setting for me to study in with the opportunity to meet and work with people with a different background than myself. Now, at the end of my research it has been a great way to end my formal education for now. During the research I had good time and learned a lot of new technologies and open source tools.

I would first thank to Ola Hodne Titlestad as my supervisor, for helping me understanding the data an assisting in the writing. I would also like to thank Jørn Braa for his valuable advice and helping me with directions on DHIS-2 development. Furthermore, I would like to thank my brothers Duong and Danh for proof reading. I could not have done it without your support and motivation.

TERESA HANG NGUYEN

TABLE OF CONTENTS

List of Figures

List of code examples

1 Introduction

In this thesis I will explore software internationalisation issues related to development and implementation of information systems in developing countries. I have done my research in the Health Information Systems Programme (HISP) where I have participated in the development of an internationalisation module and supported internationalisation efforts in the Vietnamese HISP project.

The Health Information System Programme is a research-driven collaboration between the countries of Norway, South Africa, Malawi, Mozambique, Tanzania, Ethiopia, Botswana, Nigeria, India, China, and Vietnam. HISP's overall objective is to develop and improve health information systems in developing countries through the support of information technologies. The District Health Information Software (DHIS) is developed by HISP and being implanted in larger or smaller scale in the HISP countries mentioned above.

In the open source course at the University of Oslo taken in spring 2005, I participated in the open source development of the internationalisation module for the upcoming java-based version of the DHIS, the DHIS 2.0.

In this thesis I will describe the development of this module as well as explore the challenges related to Vietnamese language support for the different versions of the DHIS software, and propose solutions on how to improve this.

Following an action research approach I will use the HISP project as a case study to understand and analyze challenges of internationalisation of information systems with a special focus on developing countries.

Research Objective

The main problem addressed is as follows:

> *To explore the importance of internationalisation in the context of information systems in developing countries, and identify challenges related to developing internationalised software in this context.*

Based on the research objective I will explore the importance of internationalisation in the information system development process of the HISP project. I have participated in the development of the internationalisation module in the future coming of DHIS-2 software and participated in the development of a general solution to enable multi-language support in the DHIS-2 software. As a developer in the HISP project I was responsible for the development of a translator program in the internationalisation WebWork-based application.

I have also studied the DHIS-1 software and HISP implementation experiences from different countries. I will explore the importance of internationalisation in HISP and look at how the HISP project enables multi-language support in the DHIS-1 software in the context of information systems in developing countries. Following an action research method I will explore experiences with adaptation to different languages in general and particularly issues related to the Vietnamese adaptation. I will study the differences in internationalisation support in the different versions of the DHIS try to identify some general challenges related

to developing internationalised software in the context of information systems in developing countries.

Structure of the thesis

Chapter 2 – presents a review of the literature used in this thesis, which includes information system in development countries and internationalisation.

Chapter 3 – presents the background, the chapter gives a short presentation of language and character sets, following with Vietnamese language and context for the research; Health Information System Programme and District Health Information Software.

Chapter 4 – introduces the methods and action research approached used with this thesis.

Chapter 5 and 6 – are related to my empirical work. Chapter 5 presents the internationalisation development of DHIS-1, my exploration findings and the challenges with translating DHIS-1 software in Vietnamese language. Chapter 6 presents the internationalisation development of DHIS-2, the investigation collaboration tools and technologies in open source software development, the translator program and interesting finding in a global open source project.

Finally, in chapter 7 my findings are discussed and conclusion is presented in chapter 8.

2 Literature Review

In this chapter I present existing literature relevant to my thesis. Theories and strategies reviewed here will help to structure the presentation of my empirical work and then lead the discussion of my empirical findings. The chapter is divided into the following sections:

2.1 – Information system in development countries

2.2 – Internationalisation

2.1 Information system in development countries

In this section I will review theories on information system (IS) relevant to my research and development project in development countries. I start with a general presentation of information system in development countries and then I look into the concept of technology transfer from north to south. At the end I look into a strategy for IS implementation called networks of action.

Challenges of IS implementation in developing countries

The traditional services and legal systems in the old days with paper-based registration had no significant IT usage. The process is often manual, slow and required a lot of time. Implementation of public information system in development countries has many serious challenges with poor record of success. Krishna and Walsham (2005) summaries the field:

> *The successful use of information technologies(IT) in the richer countries of the world has raised expectations in many poor countries that IT can be used to improve efficiency and productivity, and thereby bring about faster development in their environments. But experiences so far have often not been very encouraging, and lists of difficulties are long. A primary difficulty concerns the need of reorganize exiting systems and processes, and in a complementary way, localized processes and practices associated with the technologies.* (Krishna and Walsham, 2005)

Based on one earlier project in China, the objective was to establish a complete, operational information system for the retrieval and management of meteorological satellite data. Information and communications technologies are necessary for the processing of satellite data, but the main function of a centre for analysis of satellite data is not to invent new computing technologies or software program. This project type is to strengthen the infrastructure of activities which are vital to the welfare of developing countries. The project briefly outlines this characteristic of information technology (Baark and Heeks, 1999):

> *Most organisations, information technology functions as an "enabling technology". Access to computers and software is critical to performance, thus enabling the organisations to carry out their most important functions. However, the main objectives of these organisations typically relate to a field of activity other than IT itself.* (ibid)

In two other projects, where the first project the fundamental issue which it addressed was the effective utilisation of information technology and software in China. When the project was formulate, the task of designing better systems for input and output of Chinese characters was

seen as a major bottleneck. And the second project, the development objective was to establish a central institution to help increase the quantity of high quality software being produced in China so as to match an increasing demand for IT applications. This project type, IT-specific projects aimed directly at raising the technological capabilities of the producers of information technology in developing countries. These projects briefly outline this characteristic of information technology (ibid):

> *A few organisations, through, information technology plays a different role and, instead of being an enabling technology, it occupies as a role as a "core technology". This is true in the case of any organisation involved in computer and software development.* (ibid)

Transfer of information system

Transfer of technology and information system from the Western world (north) to developing countries (south) has been studies in several research projects and faces many problems both due to resource shortages in the developing countries and differing context in which the computer systems are installed. Too often the systems between the countries do not match the need, the organizational structures and the way work is carried out, the scarcity of resource and competence makes the adaptation of the computer system very difficult. Base on four earlier projects in China, the transfer can be conceived as five processes (Baark and Heeks, 1999):

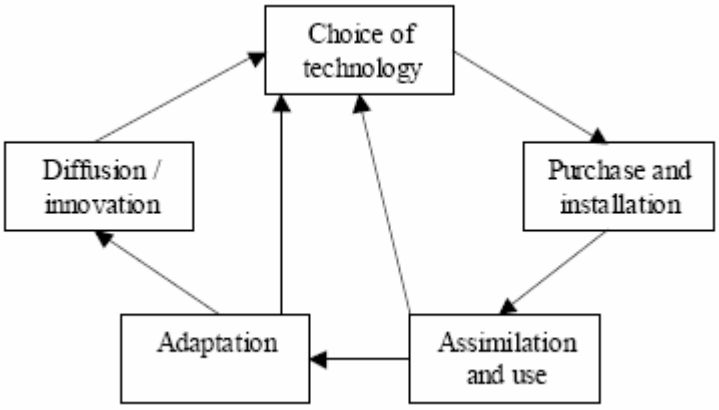

Figure 1: The information technology transfer life-cycle (Baark and Heeks, 1999)

- *Choice of technology.* During this phase the technological requirement are identified, the various alternatives for new technology surveyed, and the decision mad to purchase our otherwise obtain a particular technology. Under normal conditions this phase would be undertaken during the formulation of the project, but it may also extend into the first years of implementation.
- *Purchase and installation.* This phase involves the actual procurement of the hardware and software technology. It will often include some training and consultancy to assist in the installation.
- *Assimilation and use.* The main activity of this phase is to ensure that the people who work with the new technology fully understand how it works, how they are to use it for various purposes, and how they will maintain it on a regular basis.
- *Adaptation.* In this phase the recipient alters the technology transferred. The purpose may be to improve existing performance; to add new functions; or to match local

conditions, inputs or needs. Adaptation is frequently present, but not always (hence, the figure's line from assimilation and use direct to choice). However, only in the latter case is it seen as a fundamental part of the project, since it helps to build local IT capabilities.

- *Diffusion and/or innovation.* After the recipient organisation master the technology transferred, it can undertake diffusion to other organisations. In some cases technological innovations will be generated; production new technologies that the recipient can market locally or overseas. Even more than with adaptation, the innovation element may not always be present.

Not all projects fulfil the complete cycle, as indicated by the three arrows back to "Choice of technology".

Networks of Action

Why do so many action research projects fail to persist over time?

Based on an extensive case study in several developing countries Braa et. al have explored this question (Braa et. al 2004). They identify two main challenges to implementation of information systems in developing countries (ibid):

1. **The problem of sustainability**

   The challenge to make an information system works, in practice, over time, in a local setting. This involves shaping and adapting the systems to a given context, cultivating local learning processes, and institutionalizing routines of use that persist over time.

2. **The problem of scalability**

   The challenge of spreading and successfully adapt one working solution to other sites. The concern lies in how to reproduce and translate the necessary learning process alongside the spreading of artefact, funding, and people.

To analyze the problem of sustainability Braa et al. (2004) need to approach by an action research called *networks of action*, which described where one base action research on multiple nodes throughout a network, instead of basing it at singular sites.

> *Establishing networks creates opportunities for sharing of experience, knowledge, technology, and value between the various nodes of the experience.* (Braa et al. 2004)

Many issues about IS in development countries are common across all applications and areas. The networks of action approach address the issue of sustainability both vertically through local appropriation and linked together with horizontal processes that local routines of managing information be replicate and spread to all concern of district, provinces and countries. (ibid)

2.2 Internationalisation

In this section I present a general theories related to internationalisation. I will start with a general presentation of different internationalisation processes. Then I ended up with multi-lingual software.

> *"Internationalization is the process of designing software so that it can be adapted (localized) to various languages and regions easily, cost-effectively, and in particular without engineering changes to the software. Localization is performed by simply adding locale-specific components, such as translated text, data describing locale-specific behavior, fonts, and input methods."*(Java Internationalisation)

Sometimes the terms internationalisation is abbreviated as i18n, because there are 18 letters between the first "i" and the last "n" (ibid). Internationalisation (i18n) has several definitions in different context, from the traditional social services in everyday working life, language and culture to IT-based software, application and program across national and cultural boundaries.

In IT word, definition of internationalisation has different meaning. Based on research in Mozambique and Norway, Nielsen and Nhampossa (2005) defined internationalisation in information system as a process with the aim to reduce the costs of development support the reuse of technical and human resource and building, and sharing of knowledge across national borders. Exploring multi-lingual business software Barbour (1996) describes the key concept and definition of internationalisation as the process of making application software independent or transparent to natural language.

2.2.1 Multi-lingual Software

> *IT is being adopted by users all over the world. Software users are not limited to the English speaking west but include people from all language groups and cultures.* (Barbour, 1996)

Most early software was developed in the United States, and the demand for software was little outside these areas. As the number of computer increased around the world, demand for software also grew. With this growth in many countries especially in Asia, many software developers adapt software to other locales. The first would require is translation of software when more then one language is involved. The modification is format or display of number, data, time and character sets. If software can support more than one language, then it is internationalised. The software is fully internationalised if it supports any language. If it supports only a limited subset language then it is partially internationalised. Most translated text in internationalisation products is hard-code in the interface, as well as embedded in the language and machine operating system. Abramson believes that:

> *"The best design approach to internationalise software is to maintain strict language independence throughout the code."* (Abramson, 1994)

3 Background

This chapter presents the background for my study and empirical work. I will begin with a general presentation of languages, which describes the different character sets. I narrow down to the Vietnamese language, which is focus in this thesis. Finally, I give a presentation of the Health Information System Programme and the DHIS application in the network. The chapter is divided in the following section:

3.1 – Language and Character Sets

3.2 – Vietnamese Language

3.3 – Health Information System Programme

3.1 Language and Character Sets

Alain La Bonté says:

> *"There are around 6000 living languages on earth, 3000 almost extinct, less than a handful of which can use 7-bit US ASCII comfortably. Only 3 languages do not in practice require diacritical marks: English, Swahili and Dutch. For extended ASCII, the Latin alphabet per se, about 40 European languages can use ISO/IEC 6937, which provides roughly 330 valid characters. This list does not count extra-European languages which use the Latin script, which require extra characters, nor Vietnamese which uses extra characters too with up to 2 diacritics per character, nor Hanyu Pinyin which also use up to 2 characters per diacritic. Roughly I would say that about 50 to 60 languages in the world use the Latin script with one or more diacritical mark per character and that is between 15 and 20 % of the world population*." (Bonté, 1996)

Locales are not characterised by country alone because more than one language may be spoken in a particular country, e.g. French, German and Italian are spoken in Switzerland or multiple language are used in one country, e.g. Malay, English Mandarin and Tamil are used in Malaysia. Every language has its own alphabet or script. The 128 characters in ASCII (7-bit) code set was invented to support American English, the language of the earliest programs. The ASCII code set supported include control characters, accents, control codes and the upper and lowercase of the Roman. The Windows characters set is based on ISO standard 8859-1 also know as ECMA-94 and Latin 1. This character sets was extended to use 8-bit extended ASCII, which support 256 include characters, diacritics and umlauts that support the main languages of countries in Western and Eastern Europe, such as Cyrillic, Greek and Turkish. Languages with more than 256 characters such as Chinese, Japanese and Korean use double-byte, multi-byte or wide character sets. (Barbour, 1996)

3.1.1 Unicode support

> *"Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language."* (Unicode)

Computers store letters and other characters by assigning a number for each one. Before Unicode was invented, there were hundreds of different encoding systems for assigning these

numbers. No single encoding could contain enough characters: for example, the European Union alone requires several different encodings to cover all its languages. Even for a single language like English no single encoding was adequate for all the letters, punctuation, and technical symbols in common use. These encoding systems also conflict with one another. Two encodings can use the same number for two different characters, or use different number for the same character. (ibid)

3.2 Vietnamese Language

As Barbour (1996) mention in the previous section (ref. 3.1), there are around 3000 existing languages living on the earth. Many countries use the Latin alphabet, some other use orthographic characters and language characters in multiple countries are very different from each other. In this section I will present the Vietnamese language. I start with the Vietnamese language history and follow with the current Vietnamese written language. I continue with status of the Vietnamese language and multi-language processing, with keyboard input methods and then a presentation of the Vovisoft project.

3.2.1 History

The language is a hybrid of Mon-Khmer, Thai and Chinese elements with many of its basic words derived from the monotonic Mon-Khmer languages. In some parts one can hear several different languages are spoken like Khmer, Laotian and China. Throughout Vietnamese history and up to the present, the Vietnamese grew up speaking the dialect of the region in which they were born. There are three main dialects northern, central and southern, which correspond to three main regions of Vietnam. They differ mainly in pronunciation and in the use of some specific phrases. The Northern and Southern pronunciation are considered as official pronunciation of the Vietnamese language. (Vietnamese Online, 1997)

The official language in Vietnam is Vietnamese. The Vietnamese language is more influenced by China and has been a province of China for over 1000 years. Until the ninth century, only Chinese written language was in use. Later on some Vietnamese wanted to write their literature in their own language, they adapted Chinese character to Vietnamese in a haphazard way. This writing, called Nôm, or Chú Nôm, "*script of the spoken language"*. (JAARS Museum, 2003)

Europeans began coming to Vietnam in the sixteenth century. To learn Vietnamese, they tried writing it in Roman alphabets. A French Jesuit missionary, Alexandre de Rhodes, took these efforts and developed an efficient Roman-type alphabet for Vietnamese. Long before the days of descriptive linguistics, de Rhodes was aware of the prime importance of tone in the speech of these people. At the turn of the twentieth century, there were four writing systems in use in Vietnam: Chinese, Nôm, French, and de Rhodes'. (ibid)

In the 1940s, there was a drive toward independence and literacy for everyone in the Vietnamese language. Only the de Rhodes alphabet, called *Quốc Ngữ - "national script"*, was found suitable. Today nearly all speakers of Vietnamese are literate in Quốc Ngữ. (ibid)

3.2.2 Current Vietnamese Writing Language: Quốc Ngữ

The Vietnamese language is the current writing system and it is based on the Latin alphabet, which make Vietnam the only country using western alphabet between other countries in the region which use ideographic characters. The Vietnamese characters have 24 letters with

some digraphs and the addition of nine special marks of diacritics - four of them to create additional sounds, and other five to indicate the tone of each word. The many diacritics, often two on the same letter, make written Vietnamese easily recognizable.

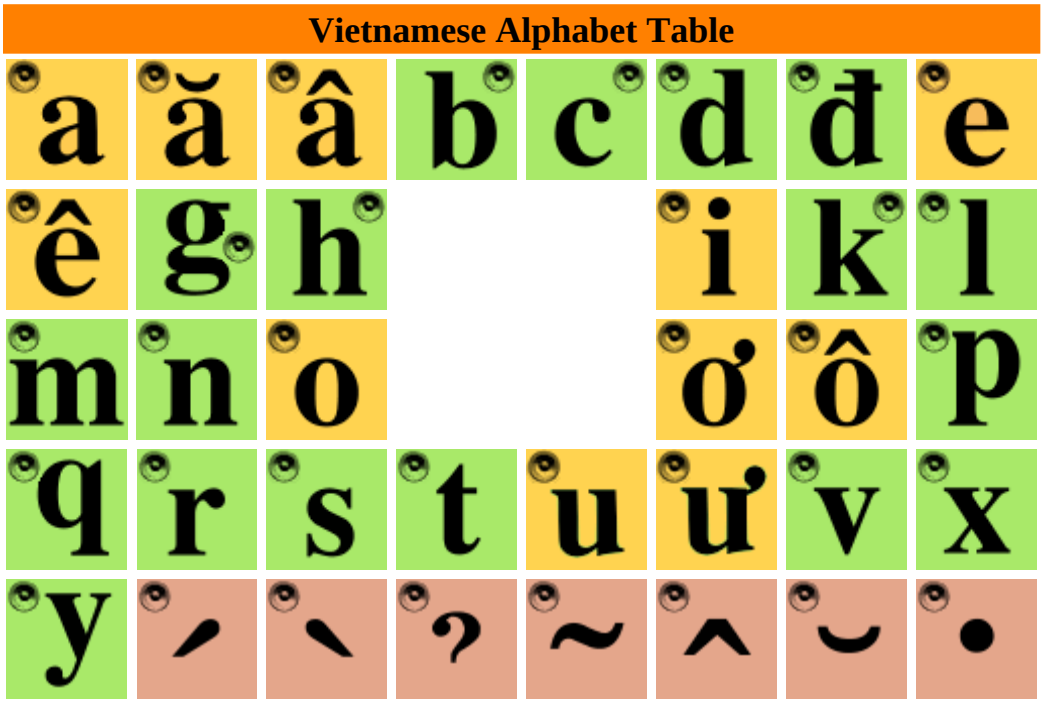

Figure 2: Vietnamese Alphabet Table (Hieu Nguyen)

The letters J, W and Z are also used, but only in foreign loan words.

Pronunciation and Tone

For speaker of English the sounds of Vietnamese are quite different. They are typically short with single consonants and one or groups of vowels. The big issue is the pronunciation of the words using the correct tone. In Vietnamese there are many words which are spelt and pronounced in exactly the same way, but meaning something quite different. The way the words are distinguished is by the tone used when pronouncing it. English usually employs tones but the tonal quality is used for emphasis or emotion.

There are six basic tones in Vietnamese, five of which are indicated by a diacritic above or below a letter, the last tone is not indicated at all, which means you keep your voice unchanged when pronouncing. (Vietnamese Online, 1997)

The word ma can have six different meanings depending on the tone:

No tone	ma		Ghost
Raising	má	'	Mother
Falling	mà	`	That
Questioning	mả	?	Tomb
Falling-raising	mã	~	horse (literally)
Weighing	mạ	.	Burgeon

Figure 3: Diacritics marks sounds (Vietnamese Online, 1997)

Vowels

The Vietnamese language uses a large numbers of vowels and groups of vowels to create different sounds, which are at first glance difficult for an English speaker to recognize. But when they are used in words, phrases or sentences, the context can help you to make them out easier.

Consonants

The Vietnamese language pronounces all kinds of consonants as single sounds. Groups of consonants are pronounced as if they were single consonants. Consonants located at the end of words are not pronounced at all, which make the language sound a bit more confusing for foreigner at the first contacts. (ibid)

3.2.3 Status of Vietnamese language and multilingual processing

Vietnam has more than 2 million people now living around the world. There is a need to use Vietnamese in communication between Vietnamese through the world. Vietnam likes to be able to exchange Vietnamese texts mixing with other languages, especially on computers and through computer networks, the Internet. (Tran et al. 1997)

Not all user interfaces of the application software is localised for Vietnamese language. Vietnam is a country with more than 50 ethnic people. Many of them have their own script for example Thai people with their Thailand script, Cham people with script like Indic script and Tay people with ideographic script, etc., and with the development of new technology, many peoples in these ethnic groups are interested to use their native language and script on computers. (ibid)

Many Vietnamese IT companies have developed and contributed their efforts in localisation of some foreign software, since 1980. At this time a national standard on Vietnamese character code set had not exist that make the situation more seriously for localisation processes. Many domestic companies and US-based Vietnamese companies made its own code table in their word processing software for its localization which where accepted and used inside the country for its localisation, and until 1991 there existed about 30 code sets for Vietnamese. The above code tables can be divided into two groups: one-byte and two byte

tables. One-byte code table is based on encoding for the extern vision glyphs, regardless on linguistic essential. Two-byte code table is based on encoding for generic parts of the language, which made the situation more chaotic and critical. (ibid)

In 1991, a task force on *Standard Vietnamese code table* had been set up by the Ministry of Science and Technology, and a result in this field for Information Interchange had been developed and after that approved in 1993 as the first national standard in IT – TCVN (3CVN). In the same year, to meet the new growing needs on IT standardization, the General Department for Standardization, Metrology and Quality Control of the Ministry of Science, Technology and Environment has established the Technical Committee on Information Technology TCVN/JTC1 responsible for IT standardisation. (ibid)

Vietnamese language processing in computers is the most interest of IT specialists in Vietnam as well as in the world. Before 1990, all proposals for Vietnamese code set are based on the pre-composed encoding scheme in 8-bit environment. And that make a lot of difficulties to keep compatible between Vietnamese and English and other requirements from 8-bit ISO standards in the Latin 1 characters set (ref. 3.1). For representing completely Vietnamese vision glyphs, 134 more pre-composed forms are needed to be encoded. The Latin 1 (8-bit) character set support 256 characters, if the set exclude upper and lowercase of the Roman, control characters, ascents, control code, diacritics and umlauts, it rests only 128 code-points for the last haft of 8-bit code set. Of course, in 16-bit environment, there is no problem because of all Vietnamese characters have been encoded in Unicode (ref. 3.1.1). A lot of discussions have been made and the conclusion is that for pre-composed encoding scheme, there is no way to keep Vietnamese compatible with ISO 8859. A local alternative can be used with pre-composed encoding scheme but it does not meet Tran et al. (1997) strategy of keeping Vietnamese in compatible with international and regional standardisation. (ibid)

In fact, there exists a bi-lingual environment for Vietnamese and English. Every application used in Vietnam must run for both Vietnamese and English. The traditional approach to this problem is to make drivers for Vietnamese (keyboard and fonts) and then to attach them into the English operating systems with a switch to turn them on / off. (ibid)

3.2.4 Keyboard - Input Methods

> *Although input of roman alphabets may be straight forward on the Qwerty keyboard, input problems arise in other languages. For example, there are more than 50,000 Hanzi characters. There exist more than 100 methods to enter these characters. Some methods to input Chinese, Japanese and Korean characters include the use of coding schemes.* (Barbour, 1996)

Vietnam has developed a national standard on Vietnamese keyboard named TCVN 6064:1995 (Tran et al. 1997), which bases on the international keyboard layout ISO 9995 and overloading some key strokes to express Vietnamese proper characters (more 6 vowels and 1 consonant, plus 5 tone marks). Beside the national standard on the keyboard layout, the most popular input methods are TELEX and VNI for *Quốc Ngữ*. (ibid)

VIQR stand for VIetnamese Quoted-Readable (VIQR, 1996) which is a convention to write Vietnamese letter using 7-bit ASCII. This convention arises from the need to exchange Vietnamese message in 7-bit environments such as email or newsgroup in the Internet. For this reason VIQR is also known as Vietnet convention. The Viet-Std Group formally

finalized the VIQR convention in 1992. As an introduction, consider this message written in the VIQR convention:

VIQR Input: To^i ye^u tie^'ng nu+o+'c to^i tu+` khi mo+'i ra ddo+`i\.

Written language: Tôi yêu tiếng nước tôi từ khi mới ra đời.

English translation: I love my language since I was born.

It can be seen that the intent is to represent Vietnamese diacritical marks as ASCII characters that suggest the shape of the original accents or tone marks. According to the VIQR standard, the following characters are subject to interpretation as Vietnamese diacritical marks:

á	à	ả	ã	ạ	â ê ô	ơ ư	ă	đ	\ /
’	`	?	~	.	^	* +	(	đ	undo

Figure 4: VIQR Input (Vovisoft d)

The VIQR standard specifies that a character is interpreted as a diacritic if and only if it can combine with the previous letter into a legal Vietnamese character. To prevent combination the character must be preceded by an escape character, which is usually the backslash (\) characters.

Later on, many Vietnam input development create theirs own font code which are indicated by a diacritic above or below a letter, as usual in the written language. As I have mention, the number of Vietnam input developments increase and today you will find around 40 code sets in the marked. A general method in this kind of input method called VNI, which you use the numbers 1-5 to mark 5 diacritics to indicate the tone and numbers 6-9 to mark additional sounds. VNI is name of software very popular in using Vietnamese on microcomputers. Because it was used for a long time and people are used to use its method to enter Vietnamese characters. Double click on one of the number buttons you will get the right number without any diacritics marks on the vowel.

á	à	ả	ã	ạ	â ê ô	ơ ư	ă	đ	\ /
1	2	3	4	5	6	7	8	9, d	undo

Figure 5: VNI Input (Vovisoft d)

When you use the touch method in the Vietnamese written language to write a document using VNI input with number buttons, the first row on the keyboard to indicate the diacritics you will slow down the work time. TELEX is an input method that had been accepted in the TCVN. It makes no change in the US keyboard layout but applying a rule in typing for making Vietnamese characters, for example: â is made by 2 consecutive typing of a. This is the third and the last popular input method to indicate the diacritics. If we change all the diacritics input from the first row on the keyboard which indicate all the numbers buttons to second, third and fourth row on the keyboard which indicate the letters buttons, it will be faster.

á	à	ả	ã	ạ	â ê ô	ơ ư	ă	đ	\ /
s	f	r	x	J	a e o	w	w	d	undo

Figure 6: Telex Input (Vovisoft d)

3.2.5 Vovisoft

The Vovisoft homepage was the first Vietnamese webpage to use the Unicode symbol to the whole homepage. This homepage is developed by a Vietnamese team living in Sydney, Australia, that help the Vietnamese people with their IT knowledge such as technical methods, software development and programming. In this site I get a lot of information related to the use of Unicode in the Vietnamese written language. There are several links that describe how to use Unicode in Visual Basic 6.0, Ms-Access, design a simple Editor using Unicode and a lot of other issue related to Vietnamese language use in JavaScript and webpage's. In this chapter I will translate two of the Vovisoft webpage's: "*History of how to convert the Vovisoft website from Vps to Unicode*" (Vovisoft a) and "*Unicode Converter develops in Visual Basic 6.0*" (Vovisoft b).

History of how to convert the Vovisoft website from Vps to Unicode

In beginning of December year 1999 webmaster of the Vovisoft Homepage visit a website in the USA. They said that they support different language in the website for the customers. But when he searches on this website, there are just some few Asia languages. He contact them and raise a question about do they support the Vietnamese font. They answer that they will support only Vietnamese with Unicode font, because all webpage's in their multi-language supported is used Unicode.

From beginning the webmaster of Vovisoft only know that Unicode is only use in some language like Chinese, Japanese, Korea, and Thailand etc. but he haven't seen any Vietnamese website using Unicode. Then he searches all the organization in Vietnam special in Input development like Vps, Visii, Vni, Vietkey etc. but he could not find anything about Unicode and how to use it in the Vietnamese written language. Even thought he gets no solution for the problem, he has a lot of resource and though in using Unicode in the Vietnamese language. In this period the Microsoft Homepage (advanced development organization support in multi-language with Unicode) have remind using Unicode in the country: Thailand, Cambodia, Lao, but nothing about using Unicode in the Vietnamese. In this period he stills this question:

> *If you are a Vietnamese, in this situation are you not angry?* (Vovisoft a)

He tried to contact his college but he got this answer:

> *"Don't bother with Unicode yet. There's no such thing yet within Microsoft. Probably, they have abandoned Vietnam because they could not make any software sales in Vietnam".* (ibid)

After a long period of searching the webmaster with his colleges explore deeper in Unicode and get a conclusion that Unicode is the only solution to solve all the problems using Vietnamese language in the webpage's. Then his colleges try to search about using the

Unicode with the Vietnamese language in the net over the whole world, articles and books and find some few parts of it.

At the end of December this year he got an email from one of his colleges with this content: *"I've changed our Vovisoft Website to Unicode tonight."* In the first two days, some Vovisoft members can read the Vietnamese font, but some other get only squares. The creator computer is using Arial font in NT4 system, but almost member with using Window 98 can not read the text. Result of the searching, the font like Arial, Tahoma and Verdana have glyphs of Vietnamese letters, but if the font size is above 170 kb or have a time stamp before June 1998 the font have not glyphs of the Vietnamese letters. After the exploration in several computer with different operative systems from Windows 95.x to 98.x, they get Tahoma is the font with the highest glyphs of Vietnamese letters. When the creator change the font type from Arial to Tahoma, he get report that 90 % of the member can now read the Vietnamese Unicode on the Vovisoft webpage.

Today they are very happy because they have gone the right direction and gone the right way. One day he using one of the computers in Internet café at Sydney, computer is using Window 98 without any fonts or software's in Vietnamese. When he visits the Vovisoft webpage with Internet Explore 5, he could read all the Vietnamese characters in the webpage. (ibid)

Unicode Convert develops in VB 6.0

Here comes my translation of *"Unicode Convert develops in VB 6.0"* from Vovisoft webpage (Vovisoft b):

Visual Basic 6.0 (VB 6.0) does not directly support Unicode. Although VB 6.0 store string internally as Unicode (UTF-16) it has several limitations:

1. Ships with ANSI only control (Label, Textbox, etc.).
2. Properties Window in IDE is ANSI only. Unicode strings are displayed as '????'
3. Property bag automatically converts Unicode strings to ANSI.
4. Clipboard functions are ANSI only.
5. Menus are ANSI only.

In this section we shall find a solution that handle and solve all the limitations above by writing a program with VB 6.0 to convert all Text files with font-codes like VNI, VPS, VISCII, TCVN etc. over to Unicode and opposite.

The first to display Unicode characters we need to use Menu command follow with the VB 6.0 IDE – Project | Components **Microsoft Form 2.0 Object Library**. This ActiveX give us all the Labels, TexBoxes, ListBoxes and ComboBox that we need to display a Vietnamese letter in Unicode. The follow, to read the Vietnamese letter in **UTF-8 Unicode**, we need to have content of all the Unicode text in a **XML file** in a tag Text (the root of the node), then use **Microsoft Document Object Model (DOM)** to read and write the Vietnamese characters. Remember Project | references Microsoft **XML, v3.0** and **Microsoft Scripting Runtime**. The whole Text is a **nodeTypedValue** root node of DOM. Doing in this way we didn't need to read each byte and find a solution to convert these data over to Unicode String. You need to use Microsoft Window NT or 2000 operation system. You can use Notepad to edit XML file with content of the Vietnamese letter and save it under the UTF-8 format.

To read and write Viet Unicode text file, we need to use VB 6.0 class clsUnicodeText:

```
Dim MyUnicodeText As clsUnicodeText
Set MyUnicodeText = New clsUnicodeText
' Read Unicode Text from file txtFileName and display in TextBox1(0)
TextBox1(0).Text = MyUnicodeText.ReadUnicode(txtFileName)
```

Listing the clsUnicodeText class:

```
Option Explicit
Private mDOMTextFile As DOMDocument  ' Document Object Model
Private mXMLPath As String  ' XML filename
Public Function ReadUnicode(TXMLPath)
' Read Unicode text from XML file
Dim objTextFileRoot As IXMLDOMElement
Set mDOMTextFile = New DOMDocument
' Remember the XML file name to update later
mXMLPath = TXMLPath
' Read the XML file and create a DOM
mDOMTextFile.Load mXMLPath
'start at the root element of the XML
Set objTextFileRoot = mDOMTextFile.documentElement
' Return the root node's text
ReadUnicode = objTextFileRoot.nodeTypedValue
End Function


Public Sub WriteUnicode(OutText, Optional TXMLPath)
' Update Unicode Text of same XML file or write to another XML file
If IsMissing(TXMLPath) Then
' Assign the Text to write out to the Root node of DOM
mDOMTextFile.documentElement.Text = OutText
' Update the XML file
mDOMTextFile.save mXMLPath
Else
' Read a dummy file to create a DOM
ReadUnicode GetLocalDirectory & "Dummy.xml"
' Assign the Text to write out to the Root node of DOM
mDOMTextFile.documentElement.Text = OutText
' Write the Text to the given XML file
mDOMTextFile.save TXMLPath
End If
End Sub
```

When using Sub WriteUnicode we can named it with one XML file if we want to save the Text in another input XML file. In this situation, the Sub WriteUnicode read one empty XML file called **Dummy.xml** to create a DOM.

All the vowels with special marks of diacritics in all encodings VPS, VNI, VISCII, TCVN etc. is store in all ANSI text file like VPSVowels.txt, VNIVowvels.txt, VISCIIVowels.txt, TCVNVowels.txt etc. And all these vowels with Vietnamese letter in Unicode is store in one XML file and will be read in the same way as all other XML Unicode text file.

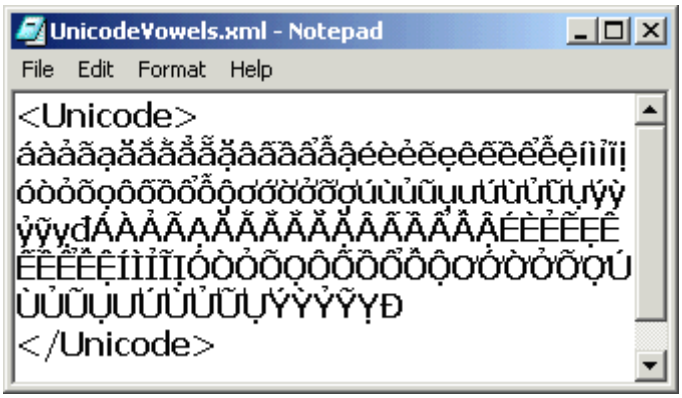

Figure 7: Unicode Vowels (Vovisoft)

When the Unicode text has been read into VB6 text string, it will be used the same way as ANSI characters. That because in the VB6 Unicode character is store openly, no matter each character needs many bytes. That mean all the functions Left, Mid, InStr can use with Unicode as same to the normal ANSI String. That's why when we convert one vowel from encoding TCVN or opposite we can use one-to-one correspondence. Let's take a look at the function StringToString to convert the vowels:

```
Function StringToString(Vowel1, Vowel2) As String
' Direct one-to-one character mapping from one encoding to another
Dim letter As String
Dim Text1 As String
Dim Text2 As String
Dim i, Pos
' Use Text1 to execute a litle faster than TextBox1(0)
Text1 = TextBox1(0).Text
' Iterate through each character of the from Text string
For i = 1 To Len(Text1)
letter = Mid(Text1, i, 1)
' Leave Carriage Return and Line Feed characters as is
If (letter = vbCr) Then
Text2 = Text2 & vbCr
ElseIf (letter = vbLf) Then
Text2 = Text2 & vbLf
Else
' Find position of character in the vowel list
Pos = InStr(Vowel1, letter)
If Pos <= 0 Then
' Not found - so do not map
Text2 = Text2 & letter
Else
' Found - so pick the corresponding character
' in the other vowel list
Text2 = Text2 & Mid(Vowel2, Pos, 1)
End If
End If
Next
StringToString = Text2
End Function
```

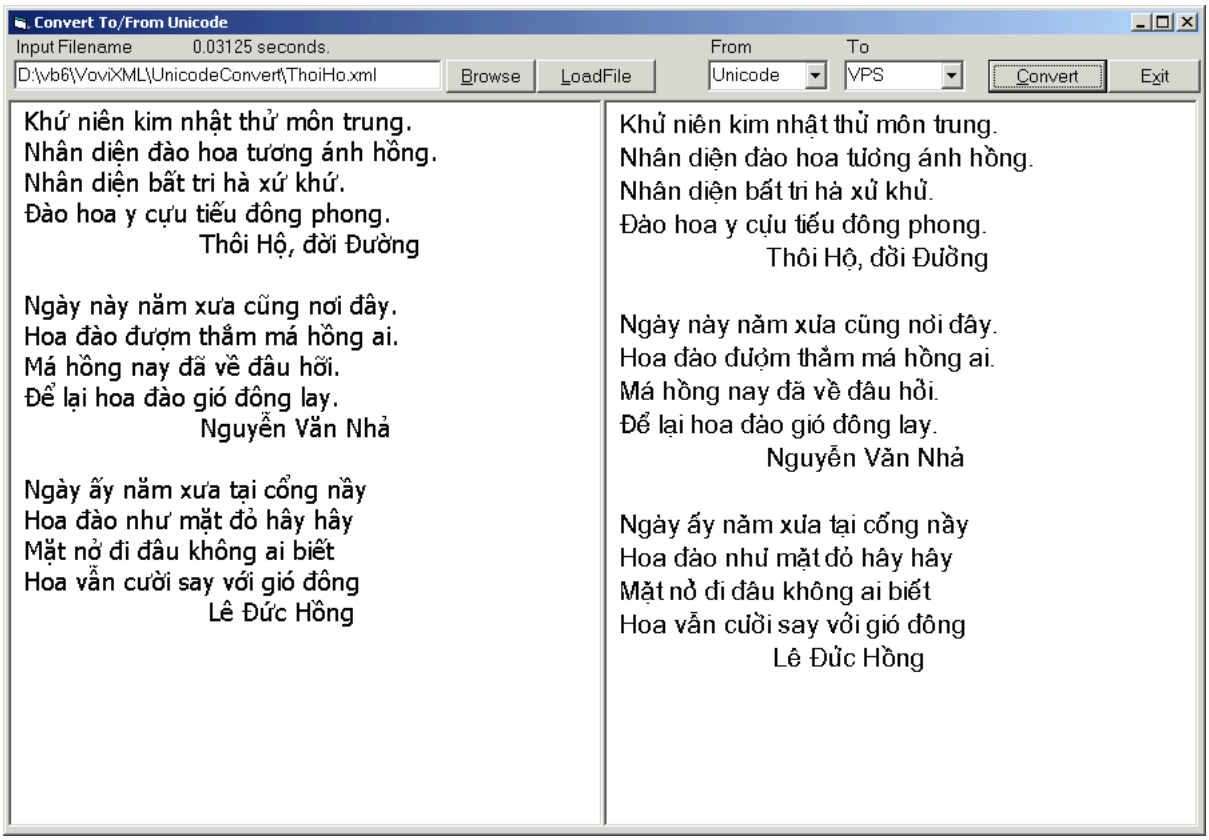

Figure 8: Convert to/from Unicode program (Vovisoft)

This program can only convert from all encodings to Unicode and opposite, but can not convert between from one encodings to another encodings witch are not Unicode.

3.3 Health Information System program

> *"The aim of the HISP is to identify information needs and to engage the community of end users and local management structures in the process of developing a new health information system."* (Braa & Hedberg, 2002)

In this section I will present the Health Information System Programme (HISP). I begin with the history and continue with goals and objectives of the HISP network. I follow up with HISP node implementation in some countries, and then a brief description of the HISP-developed software DHIS and its different versions. Finally I present previous experiences from translating the DHIS software.

3.3.1 HISP History

The HISP was initiated in South Africa as a local pilot project in three health districts in the Western Cape in 1994. The software program called District Health Information System (DHIS) was being developed by the South African team with one full-time district facilitator in each district and a project coordinator based at the University of the Western Cape. (ibid)

South Africa's Reconstruction and Development Program played a role in initiating the project. This program was one part of the development of a new national health information system, which meant a major reconstruction of the health sector in all provinces. At this time

60 % of the resources were used by the private sector, serving only 20 % of the population in the old system. Several projects where initiated, including the HISP project. HISP received funding from the Norwegian Agency for Development Co-operation (NORAD) from 1996 through 1998; a period that lead to some successful pilot projects that was later rolled out to the whole of the Western Cape. The initial HISP team consisted of University staff, activists from the health sector and nongovernmental organizations (NGOs), and two Norwegian researchers. The successful results from Western Cape lead to a spread to other provinces. The National Health Information Systems Committee for South Africa adopted the HISP processes, and the DHIS software was accepted as the national standard in 1999. (Braa and Hedberg, 2002)

3.3.2 HISP Goals and Objective

> *"The primary goal of the HISP research is to design, implement, and sustain HIS following a participatory approach to support local management of health care delivery and information flows in selected health facilities, districts, and provinces, and its further spread within and across developing counties."* (Braa et al. 2004)

The primary goal encompasses three key issues: HIS design, development and implementation, and use of information in developing countries. These goals are largely shared and it is highly contested.

The DHIS software supports local use of information and the software design process started with the following objectives (Braa and Hedberg, 2002):

- Shift of control of information systems from central towards local levels, i.e. towards more equal control between central and local levels.
- Local flexibility and user orientation – it should be easy to adapt the software to local conditions.
- Support for health sector reform towards decentralization and the development of health districts, i.e. integrating the vertical flows at district level.
- Empowerment of local management, health workers and communities.
- Horizontal flow of information and knowledge, based on the principle of free access to all anonymous, aggregated health data/information.

By standardizing information into essential datasets (EDSs) for all levels in the health sector, the DHIS software is developed to support the use and analysis of local information to aid the decision process at the health district and facilities. Braa and Hedberg (2002) made a model called *the hierarchy of standards*, which illustrates how each level have their own needs for information.

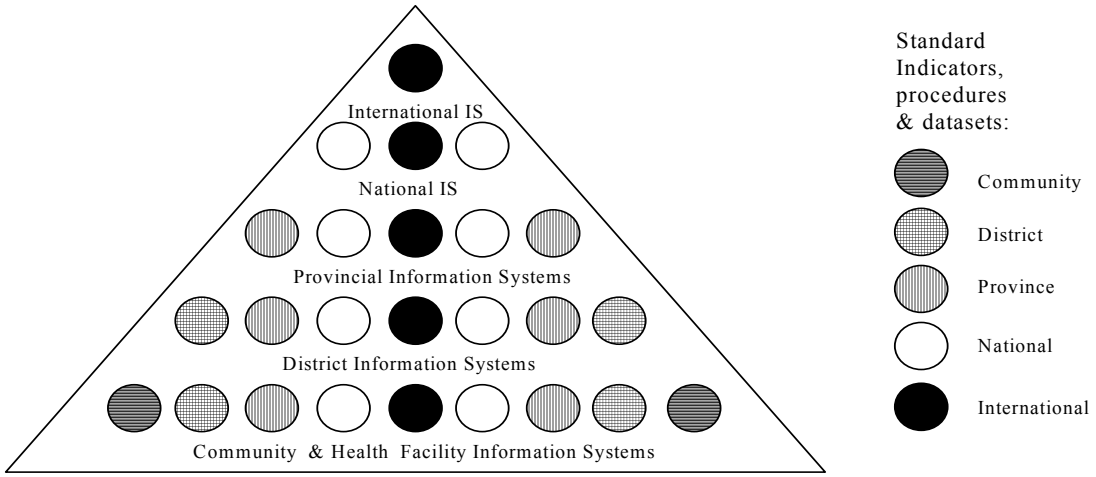

Figure 9: The hierarchy of standards (Braa & Hedberg, 2002)

Each level has the freedom to define their own standards as long as they align with the standards at the level above.

3.3.3 The global HISP project

As described earlier, the HISP project was started in South Africa in 1994 in a political climate with a shift from the apartheid regime to democracy. After about 9 months of intensive negotiation driven by local managers in collaboration with the HISP team, the first essential dataset (EDS) is implemented in all local government health facilities in the Cape Metropole, and later in the whole of the EDS in Western Cape, earlier in 1998. This process used the action research (ref. 4.1.1) in the pilot studies as a point of departure.

The HISP's initial success in South Africa in 1999 lead to the Department of Health in South Africa's adopted of DHIS as the national standard, and later to an export of HISP processes and software to other neighbouring countries. The first country was Mozambique, a neighbouring country to South Africa, which is the first node in the international network in 1999. The software has later been adopted by other countries including Tanzania, Malawi, India, Ethiopia, Botswana, Mongolia, Cuba, China and now recently Vietnam.

HISP Vietnam

Vietnam is one of the latest HISP countries where the DHIS software and the HISP strategies on health information systems are implemented. The background to the HISP project in Vietnam, HISP was contacted by the Ministry and Science and Technology and asked to do a local project in Vietnam. The interest was based in the Prime Minister's Action Plan for Applying Open Source Software and started as a pilot project in Ho Chi Minh City (HCMC), in collaboration with the HCMC Health Service. The most recent development in the project is a Memorandum of Understanding (MoU) with the Ministry of Health to develop a Free Open Source Software (FOSS) based health information system together with the Planning and Finance department (Titlestad and Staring, 2005).

The HISP project started in October 2004 and the involvement of HISP in Vietnam consists of two parallel processes, DHIS implementation and Open Source Software development. The current DHIS-1 system has been implemented in three provinces: Ho Chi Minh City (Sai Gon), Hue and Ha Noi. The focus is on establishing an integrated HIS for district and

provincial management, to improve data quality, increase data analysis and use, and to support the monitoring of Millennium Development goals (MDGs).

These dual processes have several potential benefits. First of all, the implementation of DHIS-1 provides a frame for the DHIS-2 development. The potential challenges and implication of implementation DHIS-1 can be integrated in DHIS-2. Furthermore, the DHIS-1 implementation sites provide participation of DHIS-2 development with end users that can help to evaluate the new system and give feedback on further development. Another benefit is participation of Vietnamese developers on the development of DHIS-2. This can contribute to a more locally adaptable and hence sustainable system, as it is very likely that the Vietnamese are more suited to understand the social context in which DHIS-2 shall operate than the foreign developers (Magnset, 2005).

Local participation is not only important for the development, but it is important for the global HISP community. Inclusion of new HISP countries in the development and implementation process strengthens the DHIS-2 community. With the DHIS-2 development group based in several nodes of the HISP network, one will have more sources of knowledge and experience to draw upon when DHIS-2 is to be implemented in other HIPS countries (ibid).

HISP proposal to HIS in HCMC (Titlestad and Staring, 2005):

1. Emphasis on collecting a minimum set of quality data that is of relevant to local data collectors for monitoring MDGs.
2. A focus on data analysis using an agreed minimum indicator list with defined numerators and denominators.
3. Integration of various data sources into a single database and coordination between various subsystems.
4. Training of local human resource to manage the program at all levels.
5. Support to evidence-based policy and planning processes to encourage information use for decision making.
6. Collaboration between in-country organisations (government, universities, private sector, NGOs), and donors to ensure adequate resource of Health Management Information System (HMIS) of MoH development.

The Millennium Development Goals (MDGs) for health are; reduce maternal mortality, reduce child mortality and reduce spread of special diseases.

The HISP Vietnam project is a collaboration between local universities in HCMC and Hue, both in the field of computer science and health, provincial health departments, and a local software company in Hue. (ibid):

Nong Lam University (NLU), HCMC

- Faculty of Computer Science.
- 10 students and faculties participate as software developers and support the ongoing implementation in HCMC health service.
- Collaboration with students at UiO as part of the project work in Open Source course (INF5750)
- Norwegian master students at the NLU

University Training Center for Health Care Professional (UTC), HCMC

- Dr. Cong. Head for department for community health is employed by HISP to coordinate HISP activities in Vietnam.
- Cong and two master students support the implementation, with focus on health management.

Hue University

- Collaboration with the Computer Science Faculty Computer science to let 4th year student do internships in the local HISP project, to support implementation and develop software.
- Have employed one of the graduated students that previously worked as an intern

HueCIT

- Public IT Company that develops software for the local government.
- Support implementation and software localisation.
- UiO master students have been based here when doing their fieldwork.

After one year of pilot implementation in HCMC where it started, the DHIS has been installed in all districts and city hospitals, however it is not in real use yet. Further the HISP implementation supports the Mother Care and Child Health programs. And there are some technical issues, uncertainty regarding DHIS localisation and lack of training for HISP Vietnam team, thus stronger implementation team is needed. Political brokering which struggle with HISP integration issues is included in the implementation. (ibid)

3.3.4 DHIS Software

Its core module to the DHIS-1.x modules is based on Ms-Office Office, Access and Excel Pivot tables. These modules represent the different user interfaces for monthly routine health data, client satisfaction survey and data reporting (DHIS-1.3, 2005):

- Monthly Data Module: designed to capture, validate, and process routine monthly/quarterly data, semi-permanent data, and survey/audit data usually aggregated to health facility level.

- Report Generator Module: access-based report designer with a simple GIS interface that allows the creation of health thematic maps for display in the free GIS desktop viewer ArcExplorer.

- Client Satisfaction Survey Module: access-based module for capturing and analysing Client satisfaction survey data (i.e. exit interviews with patients).

3.3.5 DHIS Versions

DHIS-1.3

The DHIS-1.3, a Microsoft Access based application, was developed 8 years ago in Cape Town South Africa and has been continuously improved since then. This version is currently used in South Africa, Malawi, Mozambique, Tanzania, Ethiopia, Nigeria, India, China,

Vietnam and Uganda with a population between 70 and 100 million people. Its core modules are based on MS Office Access and Excel Pivot tables with Monthly Data (MD), Report Generator (RG) and Client Satisfaction Survey (CSS) Module. This version covers a range of smaller stand-alone tools, most of them developed in Visual Basic (VB).

DHIS-1.4

The DHIS version 1.4 has been released in 2005 and still in development. It is important to understand that the decision to base the DHIS-1.4 to some extent on Microsoft Access was triggered by the need to design an implement major innovations that have gradually emerged over the last 5-6 years, without simultaneously have to deal with a major technology shift.

This version is regarded as the last version to predominantly rely on Office/VBA. This version should be seen as a bridging version over to DHIS-2, which will predominantly rely on the Java platform.

DHIS-2

The DHIS-2 version is a software development project in the HISP network to port the MS based DHIS software to Free and Open Source (FOSS) Java technologies. The software development process is global collaboration between students, researchers and experienced developers Norway, India, South Africa and Vietnam.

The DHIS-1.4 will largely drive the conceptual development of the DHIS the next year while DHIS-2 will focus more on implementing all those new designs and concepts in a new (Java-based) framework. The DHIS-2 in turn will probably enable and trigger another round of mainly conceptual development towards version 3.0 and so on an so forth.

3.3.6 Challenges experienced during the translation of the DHIS

As the DHIS was translated into Portuguese by Nhampossa (2004), it was tested in the pilot sites for the first time. Feedback on problems where reported to the HISP Mozambique team by the users during the piloting exercises and training session organized at various sites. Five sets of key challenges where experienced.

Language rules and lack of Portuguese equivalent terms from English

The present English language computer vocabulary was invented when the need arose when the item or concept was created (Barbour, 1996). Terms like backup, zoom, and data mart do not have direct translation in Portuguese. In this case, the team was forced to perform a partial or intermediate translation, mixing English and Portuguese text. This hybridization of terms often created problems of interpretation for the users.

The dictionary can be a good input for the translation if aligned with knowledge about concept, meaning, language rules and context of use. These meanings can be effectively provided only by people who are conversant with the language rules, cultural, context and business rules. The dictionary or the machine can only provide for translation of the standard concepts but not of the culturally specific meanings. The dictionary thus can help in the translation of simple string but not of strings to strings. Since the meanings of strings are linked to broader cultural and business understandings, the dictionary provided with the installation CD was inadequate. For example, consider the command:

```
System.out.println(string1 + string2);
```

The machine will access the corresponding text for strings 1 and 2 in the string tables and translate automatically. But if the text order has to be changed there will be a problem.

If string1 is for "yellow" and string2 is for "house" the translation for Portuguese could result in "amarela casa" instead of "casa amarela", as is the rule in Portuguese. (Nhampossa, 2004)

Length of Strings

The translation raised issues related to the length of strings as Portuguese equivalent were much longer than those used in English. This issue of length had implications for the user interfaces, the description and distribution or location of the different buttons, the layout of the screens and quality of the video adapters. Consequently, in order to keep the "correct" translation, the buttons for example had to be enlarged and located in different positions. Or the long strings had to be simplified in order to keep a reasonable layout and distributions of the buttons. For example the translation of "backup" will be *cópia de seguranca*, which did not fit in the original user interface button. In this case, the button had to be expanded with knock down implications on the design of the Menu. (ibid)

When translating text from English to another language, the Microsoft Software development Kit recommends and allowance of 30-200 percent extra space dependence on the English text length for one or more English characters. An extreme examples of expansion follow: "message pop-up" in German is "Nachrichtenuberlagerrungsfenster" Portuguese "janela de sobrepopsicao de mensagem", and in Danish, "pop-op-meddelelse" (Barbour, 1996).

Different naming convention

The Mozambique health system suffers from the problem of different and inconsistent naming conventions of the different organizational units even thought located in different provinces. For example, it is possible for a health unit in Maputo and Niassa to both have the same name of Eduardo Modlane. The naming convention thus needed to be changed, based on consultations with health authorities (Nhampossa, 2004).

Different organizational structures

The hierarchical organizational structure of the health system in Mozambique is different from South Africa. In South Africa, there are five levels including National, Province, Region, District and the Facility. In comparison, Mozambique has one less level as there are no health regions. This implied adding a dummy organizational layer to the Mozambican organizational structure to allow for the compatibility of levels (Kassbøll and Nhampossa, 2002).

Inadequate domain knowledge

Translation required computer shills, understanding of medical terminology, and knowledge of application domain and experience on translating software. The translation took place under extreme time pressure as the project needed to show quick results generated from a usable DHIS prototype to the MoH officials. For reasons of expedience, the translation of the mono-language DHIS was therefore performed focusing more on the technical terminology

and aspects from the computer point of view, rather than on developing the "correct" meanings of technical health terminology. As the English strings were hard coded, the translation took place in a traditional way, using the tools available in the Ms-Access editor (cut, copy, replace, paste). However, this technical focus led to improper meanings of terms that caused problems for the users. (Nhampossa, 2004)

Languages problems where the most critical due to lack of understanding of the terms visualized on the user interface and linked to specific functions of the DHIS. For example, the string data element was translated as *elemento de dados*, but on testing we found that the meaning was distorted by the pure text translation performed by people lacking expertise in technical health terminology. *Variável* for back the expression but this implied stopping the DHIS program and switching to design view, making the required changes, saving and restarting. In general this could happen several times during a training session leading to interruption of the training session. (ibid)

4 Methods

This chapter presents theory on relevant methods and my research approach. The chapter has the following sections:

4.1 – Research methods

4.2 – My research approach

4.1 Research methods

The research in this thesis has been done by conducted Action Research in several Case Studies. In this section I will present each of the two terms.

4.1.1 Action research

Action Research (AR) is defined as:

> *"..social research carried out by a team encompassing a professional action researcher and members of an organisation or community seeking to improve their situation. AR promotes broad participation in the research process and supports action leading to a more just or satisfying situation for the stakeholders."* (Greenwood and Levin, 1998)

AR has been typified as a way to build theory, knowledge, and practical action by engagement with the world in the context of practice itself (Braa & Hedberg, 2002). Focus on the knowledge generating process and development from participations can lead to many positive in the sense of learning. AR is a democratic method were the researchers work in active collaboration with the problem owners to solve a problem (ref. networks of action, section 2.1). There are three elements that must be present for a process to be AR:

- Research. Explore the system development in development countries.
- Participation. Local problem owners participate with the researchers.
- Action. The research must lead to something.

The role of researchers needs to be analyzed in more fined-grained manner because of two conditions. The first concerns the multiplicity, the role of the researcher has to mirror the variety in the type of actions. Several participations contribute to the HISP research efforts, in an action research mode contributing to the HISP implementation efforts the participations working on their respective thesis. The researchers do not have singular roles in HISP, but engage in district networks. The second condition is related to the political nature of the health sector. It required a minimum of support, needs to be mobilized and legitimize any changes, so it is important for the researcher team to be reasonably independent. (Braa et al. 2004)

4.1.2 Case Studies

> *"A case study is an in-depth exploration of one situation.*
> (Cornford and Smithson 1996)

This exploration often needs to have a certain time span, as a snapshot of a situation at a particular moment can not capture the processes of change. The researchers devote themselves to the specific situation, the case, and the reward is a richness of data, obtained by multiple means (ibid). Following the tradition of positivist science, a single case study can be hard to use for generalizations. By finding other such studies this problem of quality can be addressed by developing stronger and stronger evidence for certain relationships. A greater number of case studies give a better base for comparative studies.

4.2 My research approach

In this section I outline my choice of research approach. I present the team working within the project and the methods used to obtain data. I started out this project in the middle of the Open Source course taken in the spring 2005 in University of Oslo and continue with the thesis work in the same project in the fall 2005.

4.2.1 Background and aim of study

I started in this project without any knowledge about health care information system in developing countries. Since I have not so much particular knowledge of the Open Source technologies, I spend a lot of time to learn the different tools and technologies in the open source environment in the beginning of the project period. The main motivation for the thesis is to get better understanding of internationalisation of software related to information systems in developing countries.

During the action in the HISP network, I learned from my co-developers and researchers as well as through practical experiences related to the software development. This kind of training and practice was a good method to increment my knowledge, which I can hopefully build on the future.

4.2.2 HISP team

In the beginning of the project participations where divided in small team, which each team where responsible for one of the modules in DHIS-2 software. The i18n module team working in Oslo consists of the author and two additional students from the University of Oslo taken the Open Source course in the same semester. The supervisor which is responsible for the Open Source course has several years of experience from developing and implementing IS in developing countries especially in the field of health services.

4.2.3 Participation in the HISP project

My part as a member of the i18n team in the HISP project was to participate in the development of DHIS-2. The team task was to find a solution for how to multi-language the whole DHIS-2 application, create a translation program and organize the translation of the string to the supported languages of the system. In the development of the DHIS-2 software I was responsible for development of the translation program and the file control which handles reading of sentences from file to the system and opposite. Furthermore, in my research for this

thesis I have supported for the HISP Vietnam project with technical support on the translation to Vietnamese language of the different versions of the DHIS.

4.2.4 Timeline

As this thesis is of the short type, I had very strict and fixed time limit for my research. Since I come from Vietnam, the language which need the Unicode format to display letter in computer world, I have several years experience of text treatment by writing different report documents in Ms-Word before I start with the study of computer science, even thought I could nothing about programming or development of different software.

I started the project in beginning of Mars 2005 and get a brief introduction of the DHIS-1 software in two lectures. Then the HISP team investigated different open source tools and technologies around 1 week related to internationalisation before we begin with development of i18n module in DHIS-2 software. We had three month available for development of the software, at the same time I learn a lot new open source tools and technologies. At the end of the course in middle of June 2005, even we have not solved all the problems during the development of DHIS-INT, we got a first WebWork-based application running on the server with some few documentations of how to use the i18n module application. I begin with this thesis in September and continue with the HISP project in the same field. As a part of this thesis, I read a lot articles related to internationalisation before I start with the writing of the thesis. I explore DHIS-1 software especially in the structure of how to multi-language the software. At the end of the thesis I get a better understanding of internationalisation and multi-language the DHIS software in 2 versions.

5 Internationalisation of DHIS version 1

In this and next chapter I will present my empirical work. I start by giving a general presentation of internationalisation in DHIS-1 software in chapter 5, and then continue with development of internationalisation of DHIS-2 software in chapter 6. This chapter has the following sections:

5.1 – Background

5.2 – The software technologies and tools

5.3 – Challenges with translating DHIS-1 to Vietnamese

5.1 Background

In this chapter I will present the internationalisation development of DHIS-1. Since both version DHIS-1.3 and DHIS-1.4 use the same internationalisation technique, I will present these two versions in the same section.

As describe earlier (3.3.3), Mozambique is the first country with internationalisation node. The district health information software was developed using English standards, implying that the application and its documentation are written in English. In order to start using the system in the Portuguese context, the translation from English to Portuguese is required. General language competence, computer skills, and medical terminology were required in the translation, and people with these qualifications were working in the project.

Monthly Data is the main module in the health information system, and hence this module translated first. The first translation was performed focusing on the technical terminology and aspects from the software point of view, rather than health terminology in order to quickly have the first usable prototype of the system tested in the piloting sites. The English words had been hard coded in the software; therefore the translation used the tools available in the Ms-Access editor. Client Satisfaction Survey Module, the access-based module for capturing and analysing Client satisfaction survey data in front-ends (DHIS_MD.mdb in DHIS-1.3 or DHIS_CORE.mdb in DHIS-1.4). The database files, representing the back-ends of the DHIS stores information about data elements, data element categories, indicators, definitions, organizational unit data, and semi-permanent data for example related to population. For each module there is (or can be generated) a corresponding Ms-Excel spreadsheet module-pivot table that is used for data analysis. Here data can be visualized and handled in pivot tables and used to build standard or customized graphs for different purposes, for example to see the immunization coverage for different time periods. HISPML, the HISP Multilanguage library, is a separate database module storing the text strings visualized in different user-interface screens of the current front-ends. This module makes it possible to translate DHIS to all MS Windows 2000/XP supported languages, whereby by selecting the locale, the text strings are automatically visualized in the different screens. The text strings are basically stored in three categories of sources: global-dictionary, back-ends and Ms-Access forms. (Nhampossa, 2004)

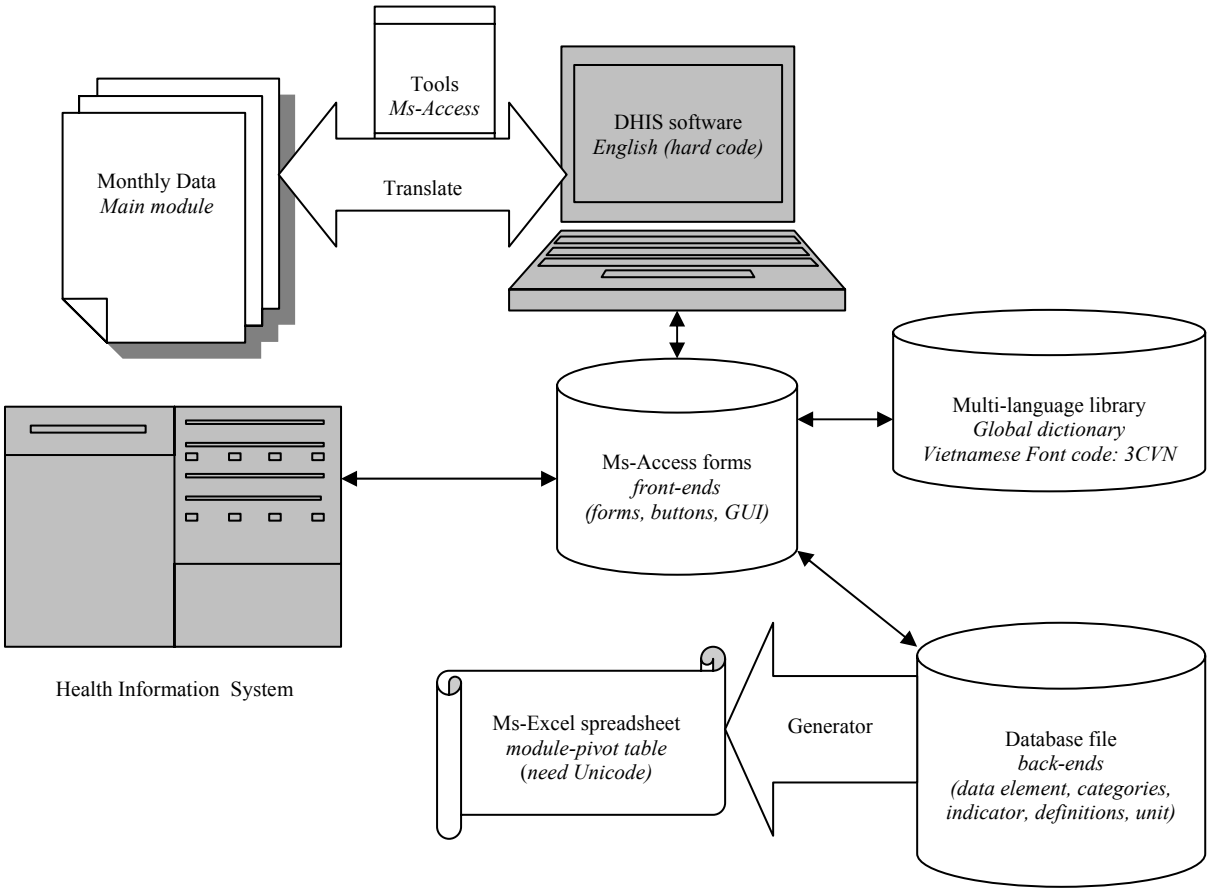

Figure 10: The HISP structure

5.2 The software technologies and tools

Regional settings

Before starting the DHIS-1 software, if you want to change the locale language in the software you need to set the requested language in the regional settings to switch from one language to another language. When the DHIS-1 software started, the current locale will be read from the region settings in the Windows system. This local will be use in the multi-language library to specify the translated text string to the current local and then automatically visualized the text strings to the requested language in different user-interface screen of the current front-ends.

Microsoft Office tools

Participations in the HISP network though that the DHIS-1 software was fully internationalised, but is seems to be its supports only a limited subset languages before the HISP project was implemented in Vietnam. The core modules in the DHIS-1.x are based on Ms-Access, Ms-Excel Pivot tables and Visual Basic 6.0. Even you can read and write Unicode characters in Ms-Excel and Ms-Access, but VB 6.0 does not directly support Unicode.

Microsoft Access

To use another font type to display all strings in CORE as opposed to the Unicode font Arial, it appears a critical problem. Because all the forms in CORE are defined with “Arial” font so

you have to redefined with the requested font type. For doing this, they have to change manually all the fonts in CORE's forms and objects. Even you can change all the fonts in forms and objects by the form name “ReLabeller” in CORE it will have a lot of resource under the changing and will leads to be time-consuming.

Windows installation system

English is the default language to every created software sales in the marked. When a software where translate from English to another language the code is still in English but the strings and the sentences view to the users display in the graphical user interface where translated to the requested language. During translating from the English to Spanish window system it appears a critical problem under the code. The Boolean data type have two variables “true” or “false” in English. Under translating of the windows installation system, these two variables where translated from the English text to the Spanish text. This leads to be an unstable running in the Spanish window system.

5.3 Challenges with translating DHIS-1 to Vietnamese

In this section I will present the challenges with translating DHIS-1 to Vietnamese. The content of this section have a lot relation with the previous section (ref 5.2).

Regional settings

Internationalisation is not fully supported in every operating system, software and tools. For example you can not find all available locales in the regional setting in every operation system. Under the Windows operating system, only XP Windows have Vietnamese in the regional settings. Since it is partly support internationalised, how can we use this DHIS-1 software in Vietnam? To solve this problem in another operative system without Vietnamese in the regional settings, you still use English in the regional settings, but you need to hard code all text in data files in the English location from English language over to the Vietnamese language.

Microsoft Office tools

You can read the Vietnamese Unicode letters in Ms-Excel and Ms-Access. Although VB 6.0 store string internally as Unicode as Unicode (UTF-16) it has several limitation (ref. 3.2.5). The DHIS-1 software use VB 6.0 to generate of new data element and transferring of data between databases e. g. export data to other databases, but VB 6.0 do not handle the Unicode characters correctly under generating of data, and many Unicode characters are displayed as ‘????’ especially in the Vietnamese Unicode character set.

Microsoft Access

The participants are struggling a lot with the Vietnamese fonts in the DHIS-1 software. Last year they found out that they couldn't use Unicode in the data files. To handle this problem in the Vietnamese locale characters, the participation found out that you had to use another font the 8-bit char set called 3CVN in order to display Vietnamese in CORE. This font code handles the characters when use VB 6.0 to generate data field and enable to display the correct Vietnamese characters in the DHIS-1 software. They use the .vnArial font which following to the 3CVN font code to display all strings in CORE, as opposed to the Unicode font Arial.

Microsoft Excel

However, Excel demands Unicode to display Vietnamese correctly in the hard-coded where they can't change the font manually. Like the drop-down list, e.g. all data element names that appears in the list when you click on the data elements field are impossible to read. Then they solved this by converting all strings in the data mart from 3CVN (8-bit) to Unicode. However, converting strings in the data mart is a difficult task as it is frequently updated with 3CVN codes strings on every "export to data mart". Then the participants ended up where they started, why is it impossible to display Vietnamese using Unicode fonts in CORE?

6 Internationalisation of DHIS version 2

In this chapter I will present the development of the internationalisation in the DHIS-2 system, the DHIS-INT. I will start with a general of the DHIS-INT module, following with requirement to the module. Then present the HISP i18n team members and the responsible field for each participant in the team. I will continue to describe how I investigate different open source software tools, technologies and development in the DHIS-INT and ended up with a presentation of the translator program. The chapter has the following sections:

6.1 – Introduction

6.2 – Requirements

6.4 – HISP i18n team and task distribution

6.4 – Collaboration tools and technologies in OSS development

6.5 – Multi-language support

6.6 – Presentation of DHIS-INT WebWork application

6.7 – Documentation and contact

6.8 – Development of DHIS-INT

6.9 – Interesting findings in a global open source project

6.1 Introduction

DHIS-INT module shall find a general solution for how to multi-language enable the whole DHIS-2 software. The language support must be complete, supporting all alphabets and languages. Furthermore, there is need for a GUI that handles translation of strings so that non-technical users can easily translate the strings. An additional task related to this module is to organise the translation of all the necessary GUI strings and documents to the languages used in the HISP network.

6.2 Requirements

The initial requirements for the DHIS-INT can be summarised in a list:

- A module for DHIS-2 which will implements i18n support
    - Translate text from one language to another
    - Other DHIS-2 modules shall be able to implement it
    - Hibernate to be used as layer between database and program module – ensure persistence

- A translator program with a Web Work GUI
    - Staff can translate text from one language to another
    - Translation shall also be read by the i18n module

6.3 HISP i18n team and task distribution

The HISP i18n team in the DHIS-INT module consists of three participations:

- **Øyvind Brucker**, responsible for designing of the GUI in the application, work on the module class, and service classes, create interface for persistence and create most of the classes in the project.
- **Eivind Austad Hovet**, responsible for creating new issues in *JIRA Issue Tracking System*, update the *Confluence Wiki* and try to implement hibernate as a layer between the database and the project.
- **Teresa Hang Nguyen** (the author), responsible for the file control class and the translation program (action classes and velocity files). This is a suitable task for me since I come from Vietnam, the language which need the Unicode support.

6.4 Collaboration tools and technologies in OOS development

Before and during the HISP project, the HISP team needs to learn some open source tools and technologies related to the DHIS software. One of the first tasks in this project was to investigate different open source technologies, related to the whole DHIS application and especially into the DHIS-INT module.

The Java programming language uses in the DHIS-2 is based on the Unicode character set. In the Java 2 Platform, internationalisation support is fully integrated into the classes and packages that provide language- or culture-dependent functionality.

6.4.1 Explore Java Internationalisation, create framework application

Our first task in the HISP team is to explore the internationalisation field from Sun Development Network (SDN a). This is a good site to visit for them who are interesting in development of internationalisation in Java. In this site you will find different problems related to internationalisation, samples in experience and solution for how to solve these problems in Java development. My first i18n framework application is based on one of the samples getting from this site.

From the Sun Development Network (SDN b), I find a sample show how the system backing a resource bundle with properties files. First of all I download the existing property files which are resource bundles (ref. 6.4.2) to each locale from the site. Each property file consist keys with correspondence values to a locale language.

```
# This is the default LabelsBundle.properties file
s1 = computer
s2 = disk
s3 = monitor
s4 = keyboard
```

Code 1: The English locale property file

```
# This is the LabelsBundle_fr.properties file
s1 = Ordinateur
s2 = Disque dur
s3 = Moniteur
s4 = Clavier
```

Code 2: The French locale property file

I continue to design the framework window with labels and textboxes represent the keys and values (computer, disk, monitor and keyboard) getting from the property files following with buttons related to each locale language. Then I create an action perform function, which get the locale, when action from the user by click on a specific language button. The system will get content of the values based on the keys in the requested locale property file and put these values into the correspondences textboxes in the framework. In combination with the sample above (SDN b) and the Java framework, I created my first framework application in Java swing with resource bundle in three languages: English, French and German. The application will change the string on textboxes from one language to another language when the user clicks on one of the language buttons.

Figure 11: Java framework application

Later on I get an email from one of the developers from Reporter Designer module which tell that she could not use the Vietnamese language with Unicode character in her framework application (will describe in more detail in Unicode support, section 6.9.2). And addition to this framework application based on the Vietnamese encoding table with Unicode (ref. Appendix B, section 10.2), I add the Vietnamese resource bundle and create the objects which follow the Vietnamese language and the framework application are now supported with four languages.

```
# This is the LabelsBundle_vi.properties file
s1 = máy \u0111i\u1EC7n toán              -> máy điện toán
s2 = \u0111\u0129a                        -> đĩa
s3 = màn h\u00ECnh                        -> màn hình
s4 = bàn vi\u1EBFt ch\u1EEF               -> bàn viết chữ
```

Code 3: The Vietnamese locale property file

6.4.2 Resource bundle

All words and sentences in the DHIS-2 software both under Ms-Office software and webpage's are hard-coded. To internationalise the application we need to rebuild theses sentence by converted these sentences from static pages (hard-code) to dynamic pages by

store these sentences in property files, one file for each supported language called resource bundle.

The class Resource Bundle is an abstract base class representing containers of resources. Programmers create subclasses of resource bundle that contain resources for a particular locale. New resources can be added to an instance of resource bundle, or new instances of resource bundle can be added to a system without affecting the code that uses them. Packaging resources as classes allows developers to take advantage of Java's class loading mechanism to find resources. Resource bundles contain locale-specific objects. When a program needs a locale-specific resource, a String object for example, the program can load it from the resource bundle that is appropriate for the current user's locale. In this way, the programmer can write code that is largely independent of the user's locale isolating most, if not all, of the locale-specific information in resource bundles. This allows Java programmers to write code that can:

- be easily localized, or translated, into different languages
- handle multiple locales at once
- be easily modified later to support even more locales

All the words/sentences to a locale should be saved in

```
DHIS2MessageBundle_locale.properties file.
```

This is the resource bundle with content of all keys and corresponding words/sentences (values) to a locale in DHIS-2 application. English (UK) is set to the default locale and will be supported in the system. File control class should be used to get content of the selected locale in the resource bundle as hash table through the global service class. This hash table will be use to get the keys with values to the current locale and display it in the I18N module Web Work application.

Resource bundle is a property file to a specific language consist of keys (identify the translated text) and values (translated text). The name of these file should be on this format:

```
DHISMessageBundle_vi_VN.properties
```

The first part of the file name “DHISMessageBundle” you can choose yourself. The second part is reserved and indicates the country code. The third part is reserved and indicates the language code. This mean that the format above is a Java resource bundle file named DHISMessageBundle which contains text strings for Vietnam (vi) country code in the Vietnamese (VN) language code. When you use the default (English) resource bundle to get a sentence in the English language based on a specific key, you will find the translated sentence with the same key in another language (Vietnamese).

6.4.3 Hash table

Hash table is an existing defined class in the Java library. We use this hash table in the DHIS-2 software to find the translated string to a locale based on a key. A hash table exist containing following resources for locales.

1. A key that identify each of the translated string
2. A string contains the translation for this key for the specific locale.

This hash table is a data structure, works similar as a dynamic array, and the methods using in the software following the class is easy to use. Put method put a key and a value to the hash table. PutAll method put all keys and values from another hash table into the current hash table. Get method get value in the hash table based on a given key. And remove method remove the key with the corresponding value from the hash table.

```
hash.put(key,value);
hash.putAll(secondHash);
hash.get(key);
hash.remove(key);
```

Code 1: Hash table methods

6.4.4 Web Work as interface

Our group and all the modules in DHIS-2 application which develop a web application should use WebWork as interface. This is another general user interface related to the web application. We have focused on Web Work integration with Java and tried to make an alpha version with successfully result.

6.5 Multi-language support

The objective of the i18n module is to internationalise the DHIS-2 project. A major part of this objective is to handle translation of string for every locale the application will run at.

How the module works

The DHIS-INT module gets a translated string for a given locale based on a key that identifies this string; English (UK) is used for the naming of these keys. English (UK) is also the default locale to use if the necessary information cannot be obtained for the current locale.

To multi-language the application I create a start page based on the index page Øyvind had created before. I make a dropdown list to get supported locales in the system, when the user selects a locale, the work locale will be set in the session and all the string on the webpage will be translated to the selected locale language. When the next page execute I get the current work locale from this session and use this locale to get content of the resource bundle as hash table. When this task have been solved I change all my velocity files from static text over to dynamic text by getting the sentence to each string in the webpage and using the hash table to get currently English work locale resource bundle. Then make resource bundle to the Norwegian and the Vietnamese language and the i18n module WebWork application will now be supported with 3 languages. Development of this task will be describes under communication and implementation with other DHIS-2 software modules (ref. 6.8.1).

6.6 Presentation of DHIS-INT WebWork application

On of the task in the i18n module HISP team is developed a GUI Web Work application. The web application has five main pages with a menu which is available on top of each page.

- Start - In the start page the user can select the currently work locale in a dropdown box. Content of this dropdown box is supported languages in the system. The work locale will be set to the default locale English (UK) in the session if the user has not selects any other locale. This work locale will be reflected to the whole application.

- Select locales - This page get all available locales with corresponding language and country in the system and display to the webpage. The user can choose which language to view text to be translated (reference locale 1 & 2) and a requested language to translate text into (locale). When a locale is selected the user will be transfer to the translator webpage. To select an optional reference locale the user has to go back to this page and get a new selection.
- Translator - The main translation program.
- Setup - A little setup system with key management. The page is created for developers in all other modules in DHIS-2 application which support internationalisation. The user can add new key and value, update a value in a specific key or delete a key in the resource bundle to the default locale English (UK).
- Help - A little help system with questions and answers that help the user with the i18n module Web Work application.

File and folder structure to DHIS-INT application is presented in Appendix A (ref 10.1).

6.7 Documentation and contact

Our goal was to inform staff using DHIS-2 about the translation of documentation. We wanted to keep contact with developers in charge of language and translation in different countries. Sadly, this did fail for some reason. We had contact with foreign developers, but not in the documentation sense. It was strictly programming. Our platoon should gather information and get an overview of necessary translation of documents.

6.8 Development of DHIS-INT

One of the DHIS-INT module tasks is to develop a translation program. This program should be a graphical user interface where non-technical language responsible users can translate words/sentences for every locale in DHIS-2 application. The look of the program has to be resizable. When a given language demands more space than the English text, the GUI will resize to fit the new dimension. In this section I will describes how I use the different technologies and tools related to the DHIS software to develop the DHIS-INT WebWork-based application and then present how the DHIS-INT application works.

6.8.1 Communication and implementation with other DHIS-2 modules

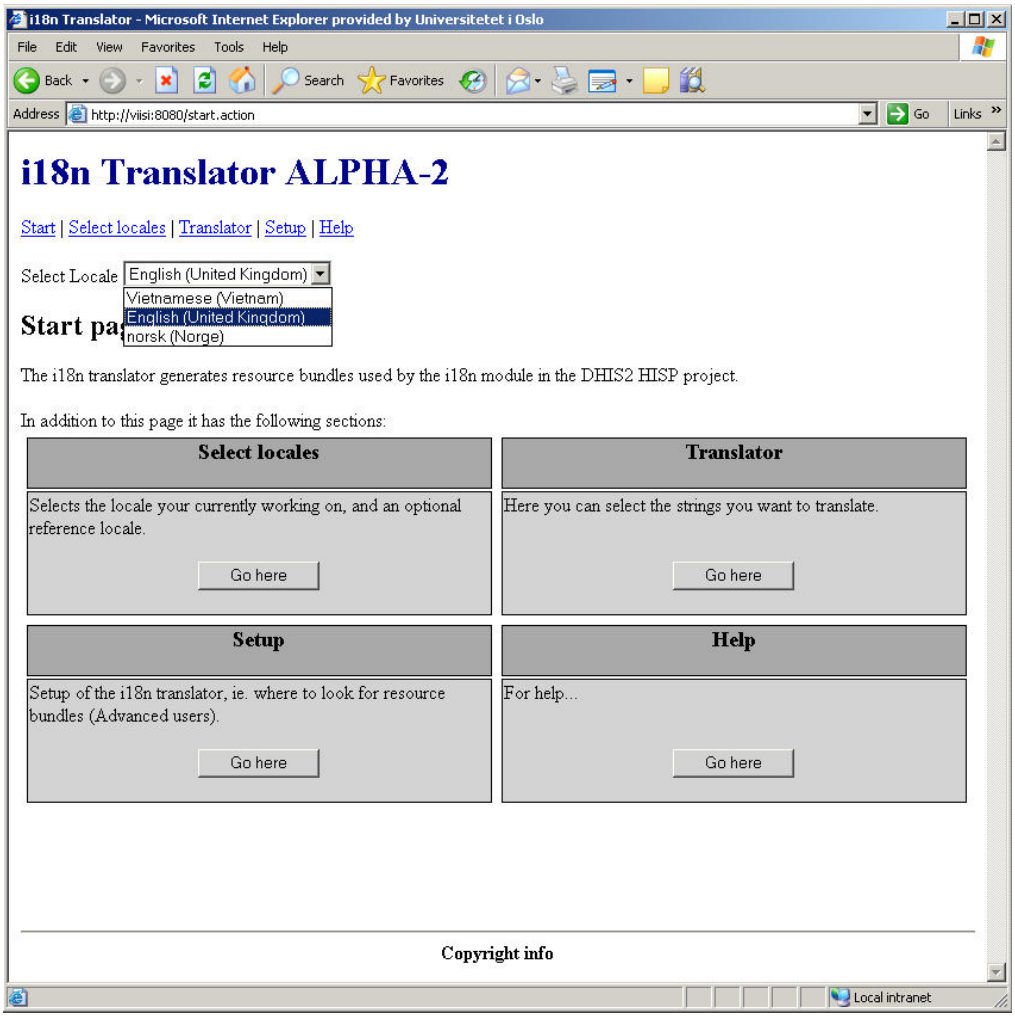

Figure 12: Start page

Our module has to communicate with other modules that are dependent of us. This sub-section gives you an instruction of how other implement to our module.

Every webpage is usual mapped to a class *action.java. In the start page when you open the web application, you need getLoc/setLoc methods to get/set the current work locale. In the execute method you need to call setLoc(getLoc().toString()) to set the default work locale in this session.

```
public Locale getLoc() {
  /*
  *  Return the current work locale from this session. If the locale
  *  is null return the default work locale in this session
  */
  Locale locale = (Locale)
ActionContext.getContext().getSession().get("workLocale");
  if(locale == null)
    return globalService.getDefaultLocale();
  else
    return locale;
}

public void setLoc(String locale) {
  /*
   *  Using private method makeLoc to create a Locale object based on
   *  the input string. Set as the reference locale in this session
   */
   ActionContext.getContext().getSession().put("workLocale",
globalService.makeLoc(locale));
}
```

Code 4: Get and set locale methods in startAction class

To select the locale you currently working on, I have done a dropdown list to get all supported locales in the system. When you select a locale, the work locale will be set in the session and all the string on the webpage will be translated to the selected locale language.

```
<form name="Loc" action="start.action" method="post">
$locHash.get("select_locale")
  <!-- Select the local property file you want to work -->
  <select name="Loc" onChange="document.forms.Loc.submit();">
    #foreach ( $locale in $existingLocales )
      #if( $locale == $loc )
        <option selected value="$locale">
            $locale.getDisplayName()
        </option>
      #else
        <option value="$locale">$locale.getDisplayName()</option>
      #end
    #end
  </select>
</form>
```

Code 5: Drop-down list in startAction class

You use file control class to get all supported locales and to get content of the current work locale property file as hash table. Default work directory method in the file control class use to browse to the property file is set to

```
Your root directory + "src/main/resources/ResourceBundles/global/"
```

To use this file control class from the DHIS-INT module you need to set new work directory before using the methods in the file control class.

```
FileControl fileControl = new FileControl();
fileControl.setWorkdir(
"../dhis-int/src/main/resources/ResourceBundles/global/");
Locale[] existingLocales = fileControl.getExistingLocales();
Hashtable locHash = fileControl.getHash(workLocale);
```

In the resource bundle file to every country, you get the keys language, country and help with corresponding values in English (en_GB), Norwegian (no_NO) and Vietnamese (vi_VN):

```
language = Language#¤¤¤#
country = Country#¤¤¤#
help = Help#¤¤¤#

language = Språk#¤¤¤#
country = Land#¤¤¤#
help = Hjelp#¤¤¤#

language = Ngôn ng&#7919;#¤¤¤#
country = N&#432;&#7899;c#¤¤¤#
help = Giúp &#273;&#7905;#¤¤¤#
```

From the velocity files you use the keys to get their corresponding values in the resource bundle files:

```
<table>
  <tr><td>English</td>Norwegian</td><td>Vietnamese</td></tr>
  <tr><td>language</td><td>$locHash.get("language")</td></tr>
  <tr><td>country</td><td>$locHash.get("country")</td></tr>
  <tr><td>help</td><td>$locHash.get("help")</td>
<table>
```

In the webpage all the $locHash.get(“key”) will be translated to their current work locale values.

	English	**Norwegian**	**Vietnamese**
language	Language	Språk	Ngôn ngữ
Country	Country	Land	Nước
Help	Help	Hjelp	Giúp đỡ

Figure 13: Multi-language table

When next page execute, you need to get the current work locale from the session. Then you use this locale to get content of the current property file as hash table.

```
Locale workLocale = (Locale)
ActionContext.getContext().getSession().get("workLocale");
FileControl fileControl = new FileControl();
fileControl.setWorkdir("../dhis-
int/src/main/resources/ResourceBundles/global/");
Hashtable workLocHash = fileControl.getLocaleHash(workLocale);
```

You are ready to use this hash table in your velocity file.

6.8.2 Get and select available locales

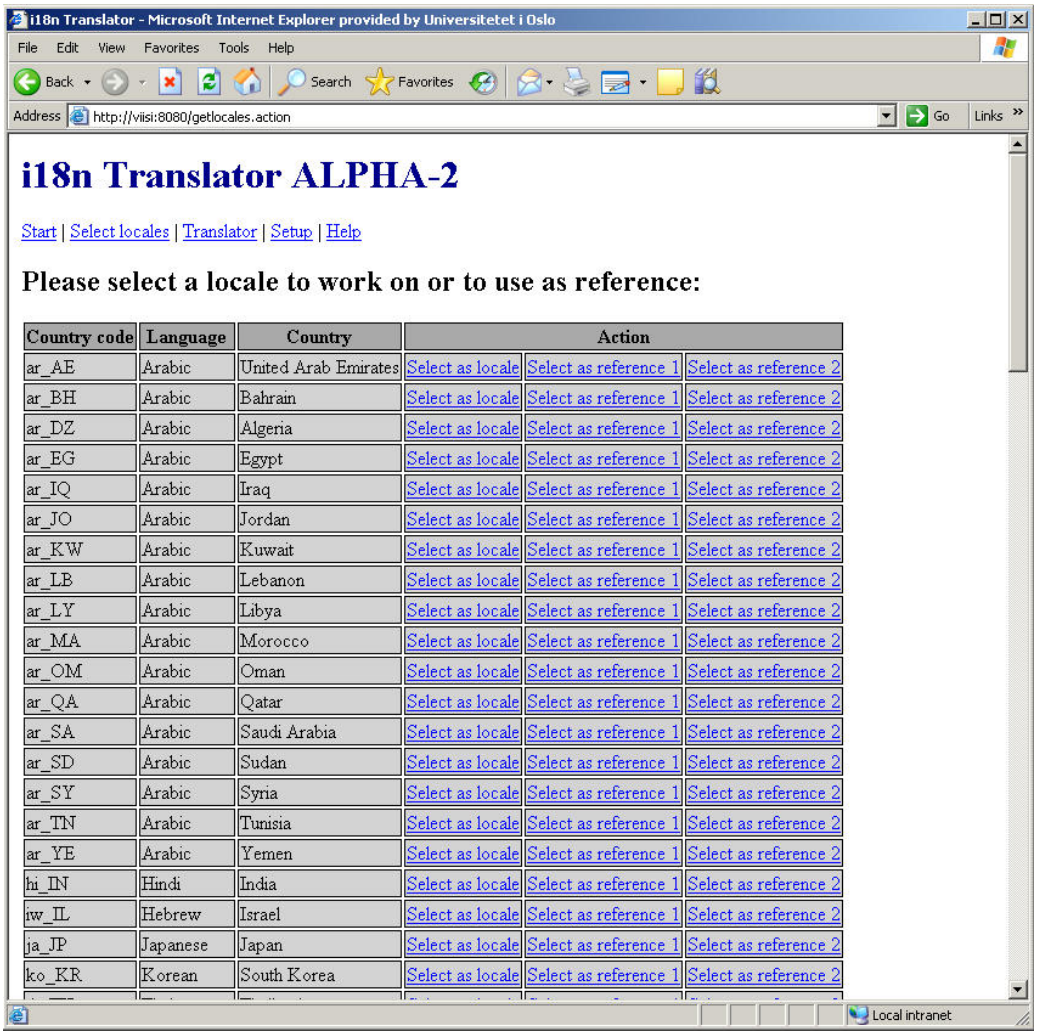

Figure 14: Select locale page

Content of this page is all available locales with corresponding language and country. Locale is a combination of valid languages and country code (SDN c). To find out which types of Locale definitions a locale-sensitive class recognizes, I invoke the getAvailableLocales method. As describe in the language and character sets section (ref. 3.1), locales are not characterized by country alone because more than one language may be spoken in a particular country or many languages are used in one country. One locale has even only country code as default, or has both country code and language. The first part identifies the language and the second identifies the country code. The following examples explain the situation.

```
en English
en_AU English (Australia)
en_CA English (Canada)
en_GB English (United Kingdom)
en_IE English (Ireland)
en_IN English (India)
en_NZ English (New Zealand)
en_ZA English (South Africa)
```

Code 6: English language in different country

To get locales with both country code and language, I create a new empty locale array. Then put all available locales to the array if the locale to each country exists.

```
public Locale[] getLocales() {
/*
 * Returns available locales with country and language.
 * Use a vector class to delete the locale if no country name exists
 */
  Locale[] locales = Locale.getAvailableLocales();
  Vector country = new Vector();
  for(int i=0; i<locales.length; i++){

    // Set the locale to the vector
    // if the current locale country exists
    if(locales[i].getCountry() != ""){
      country.add(locales[i]);
    }
  }

  // Convert the vector back to the locales array
  locales = new Locale[country.size()];
  for(int i=0; i<country.size(); i++){
    locales[i] = (Locale) country.get(i);
  }

        // Content of the locales will now have
  // both country and language
  return locales;
}
```

Code 7: Get locales method in localeAction class

6.8.3 *Translation program*

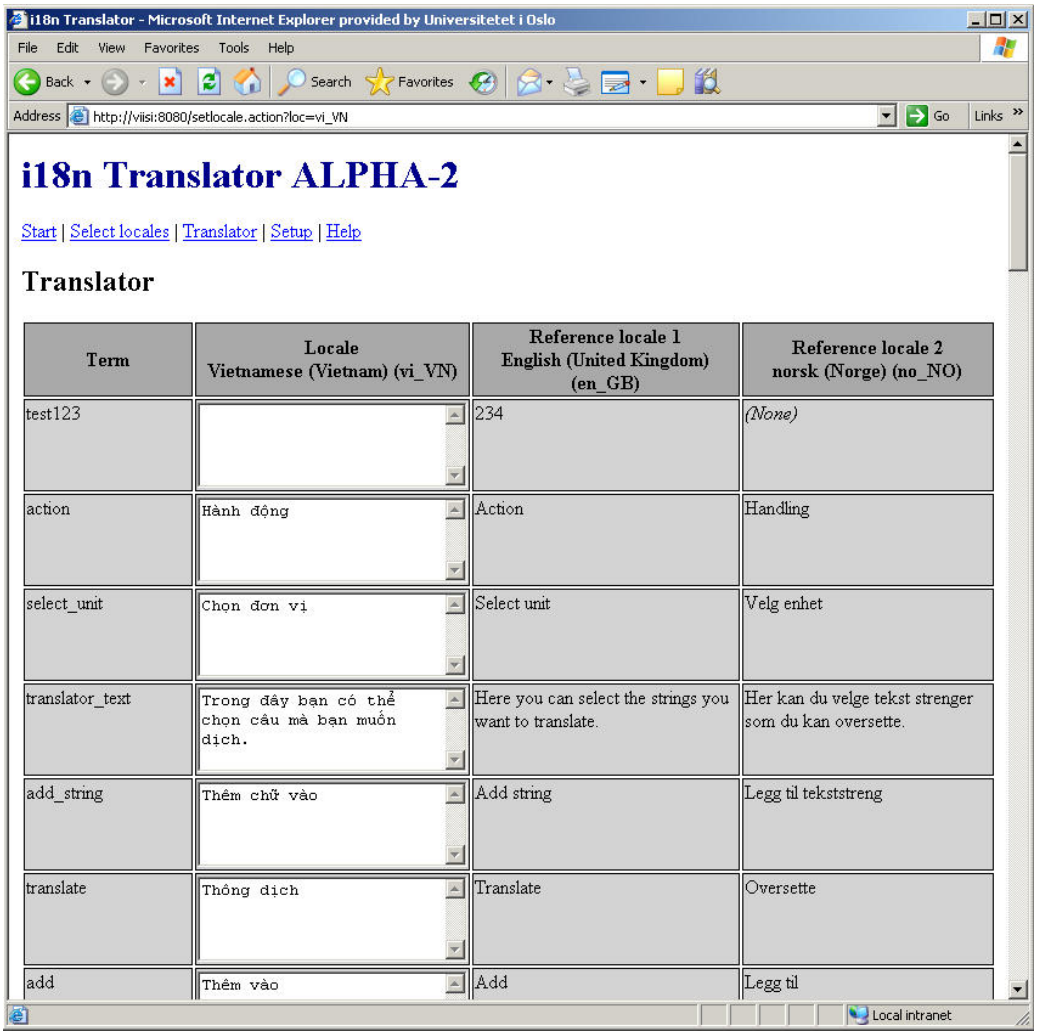

Figure 15: Translator page

The main task in the 18n module is to develop a translation program. This program should be a graphical user interface where non-technical language responsible users can translate words/sentences for every locale in DHIS 2 application. The look of the program has to be resizable. When a given language demands more space than the English text, the GUI will resize to fit the new dimension.

In this sub-section I will explain how to use the translation program. The program will display both translation of text and the original text (which was translated). Content of this page is a table with four columns:

- Term - Represent the keys in the resource bundle where the system use to identify the translated text, they are also available for the reference locale.

- Locale - Content of text area to each key where the user can translate the words/sentences from the selected reference locale to a requested language. If the current locale doesn't support in the system, all text area will be empty. If not the words in the locale language should display in text area and the users can edit/update the words/sentences to the current locale.

- Reference locale 1 - Display the words/sentences to help the users with the translation in DHIS-2 application.

- Reference locale 2 - A second reference locale if the users need to have two reference languages to translate the requested language.

When all the words/sentences are translated to the selected locale, the user click on the translate button on the bottom at the page and the translation words/sentences will be save to the current locale resource bundle in the system. The user will be transfer to the translate webpage where the system give feedback that the current file is save successfully. Content of the saved file with keys and corresponding translated words/sentences will display to the page. This new locale will now be supported in the system.

6.8.4 Figure out how to pair keys with corresponding translated text in the translator

The translator webpage is the translation program, and this is one of the most difficult tasks in the I18N module. As describe earlier, the first columns represent the keys in the resource bundle and the second represent text area to each key where the user can translate the words/sentences from the selected reference locale 1 & 2 to a requested language.

```
<table>
  <tr>
    <th>Term</th>
    <th>Locale</th>
    <th>Reference locale 1</th>
    <th>Reference locale 2</th>
  </tr>
  <tr>
    <form action="save.action" method="post">
      // Use a for-loop to list the keys and values
      // from the property file
    </form>
  </tr>
</table>
```

Setter and getter method

The following example by using setter and getter method with the result from the test describes the problem if the user used comma in the translated sentence.

First of all, I get content of the keys and words/sentences in the reference locale as hash table from the file control and use a for-loop to display it to the webpage.

Key	**Value**
Hi	Hello
Morning	Morning, sir
Night	Good night
Bye	Bye bye

Then I get content of keys and values columns from the web page and set the keys to the variable name=’key’ and the translated text areas to the variable name=’translate’. When I click on the translate button (submit) on the bottom of the page all the keys and translated sentences will sent to next page as two separate text string with comma.

Key	Hi, Morning, Night, Bye
Translate	Hello, Morning, sir, God night, bye bye

Since the system use comma to separate each element (key and value) in the text string, the user can not use comma in the translated sentences. The result from this test, the key doesn’t match with the value and resource bundle loose the last even more value(s) if both the user and the system used the comma in the translated sentences and to separate each element.

Key	**Value**
Hi	Hello
Morning	Morning
Night	sir
Bye	Good night

I try in several days to solve this task with coma supported in the translated text by using JavaScript, ask friend for helping and write in the mailing list but get no answer and solution for the problem. After telling responsible of the group assignment in the open source course about the problem, he answer that the Standard Data Entry module had solved this problem. I ask them for help and one of the participants in this module shows me how they solved the problem.

The CGI parameters implement with Parameter Aware

As a result I use velocity count from the velocity file *translator.vm* to set the keys and translated text as two arrays.

```
#foreach ( $key in $allHash.keySet() )
  #set ( $index = $velocityCount - 1 )
  <input type="hidden" name="key[$index]" value="$key">
  <textarea name="translate[$index]"></textarea>
#end
```

In the saveAcion class the system receives CGI parameters wrapped in a map and must therefore implement Parameter Aware. The alternative method using setters and getters would not work because the CGI parameters are arrays which can be of different lengths. In the execute() method I use a for-statement to loop through the arrays and collect associated CGI parameters and put the keys and values to the hash table and save it to the current locale property file (resource bundle) in the system.

```
public class SaveAction
  extends ActionSupport
  implements ParameterAware {

  public String execute() throws Exception {
    for(int i=0; i<ArraySize; i++){
      // Loop through the keys and translate arrays
      // and collect associated CGI parameters
      String[] kid = (String[]) parameters.get("key["+i+"]" );
      key[i] = kid[0];
      String[] tid = (String[]) parameters.get("translate["+i+"]" );
      value[i] = tid[0];
      // Put the keys and values to the hashtable
      hash.put(key[i], value[i]);
    }
    // Save the hashtable to the current locale resource bundle
    globalService.saveHash(locale, hash);
    return SUCCESS;
  }
}
```

Code 8: Execute method in saveAction class

6.8.5 Setup system, add key management to Web Work GUI

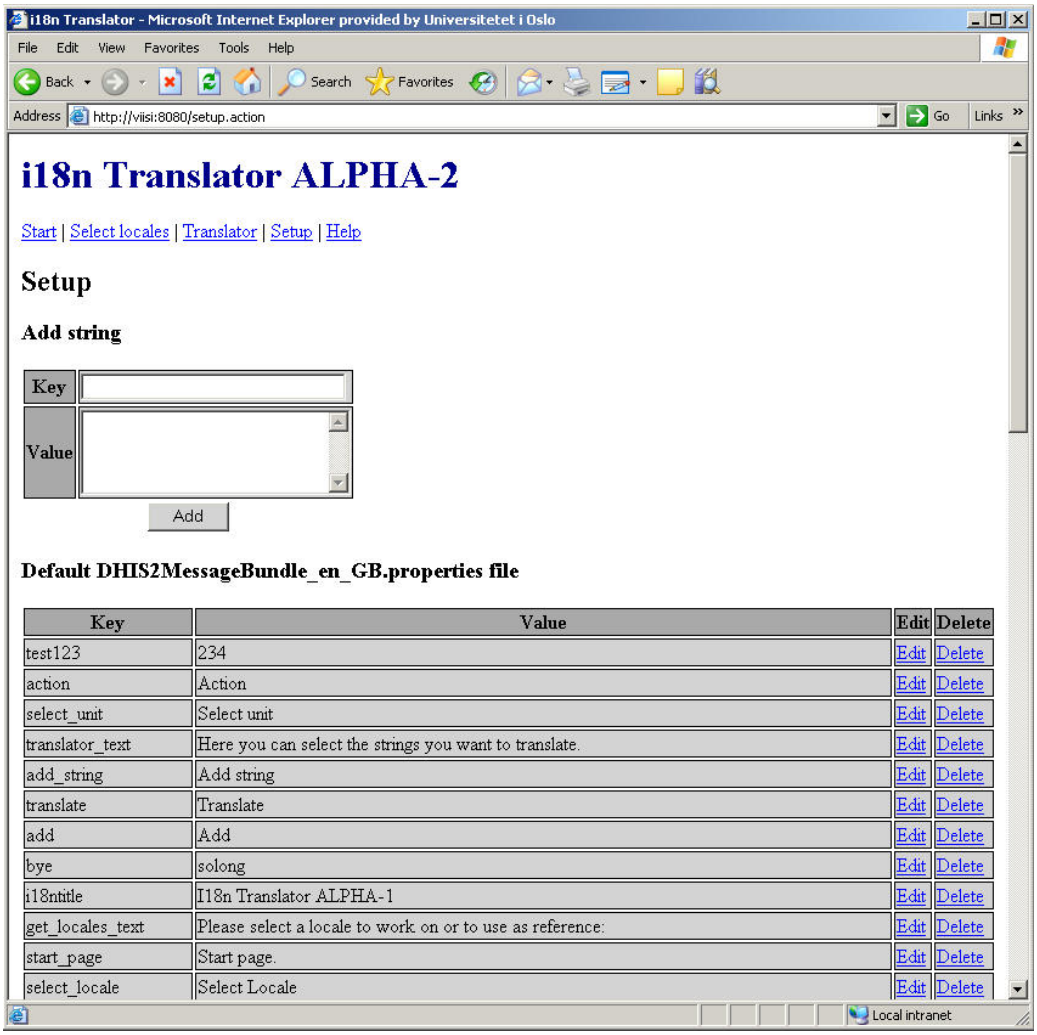

Figure 16: Setup page

New modules appear in the futures and existing modules which support internationalisation need to add new keys and strings to the resource bundle. The application needs a little setup to management all the locale keys and values in the system. My task in this sub-section is to add key management in the setup webpage.

First part of the setup webpage is a form where the user can add or update a key, when the user edit to the text box represent the key and text area represent the value and click on the submit button, the save hash method in the file control will be used, new string will be saved and will be available in the default English (UK) locale resource bundle.

```
<h2>Setup</h2>
<form action="messagebundle.action?select=add" method="post">
<h3>Add string</h3>
<table class=normal>
  <tr>
    <th>Key</th>
    <td><input type="text" name="key" size="32"/></td>
  </tr>
  <tr>
    <th>Value</th>
    <td>
      <textarea name="value" type=text rows=4 cols=25></textarea>
    </td>
  </tr>
  <tr><td colspan="2"><center>
    <input type="submit" value="Add" class="button"/>
  </center></td></tr>
 </table>
</form>
```

Code 9: Setup page, first part

Next part is a table which displays all the keys with corresponding values to the default locale resource bundle. In this table you will find two links to each key, ‘Edit’ and ‘Delete’. The edit links set the selected key and value to the form at the top of the page where the user can set/update new value to the selected key to the default locale resource bundle in the system. The delete link deletes the selected key in all supported resource bundle in the system.

```
<h3>Default DHIS2MessageBundle_en_GB.properties file</h3>
<table class=normal>
  <tr>
    <th>Key</th><th>Value</th><th>Edit</th><th>Delete</th>
  </tr>
#foreach ( $key in $locHash.keySet() )
  <tr>
    <td>$key</td>
    <td>$locHash.get($key)</td>
    <td><a href="messagebundle.action?select=edit&
          key=$key&value=$locHash.get($key)">Edit</a></td>
    <td><a href="messagebundle.action?
      select=delete&key=$key">Delete</a></td>
  </tr>
#end
</table>
```

Code 10: Setup page, second part

An addition to this webpage I set a validation function in JavaScript to validate that the key and the value text box should not be empty when you add a new key.

```
<!-- validation function -->
function validate() {
  x=document.addBundle
  submitOK = "True"
  text = ""

  if (x.key.value == ""){
    text = "Please enter a key"
    submitOK = "False"
  }

  if (x.value.value == ""){
    if (text == ""){
      text = "Please enter a value"
    }
    else{
      text = text + " and a value"
    }
    submitOK = "False"
  }

  if (submitOK == "True"){
    return true
  }
  else{
    alert(text)
    return false
  }
}
```

Code 11: Setup page, validate function

6.8.6 *Help system*

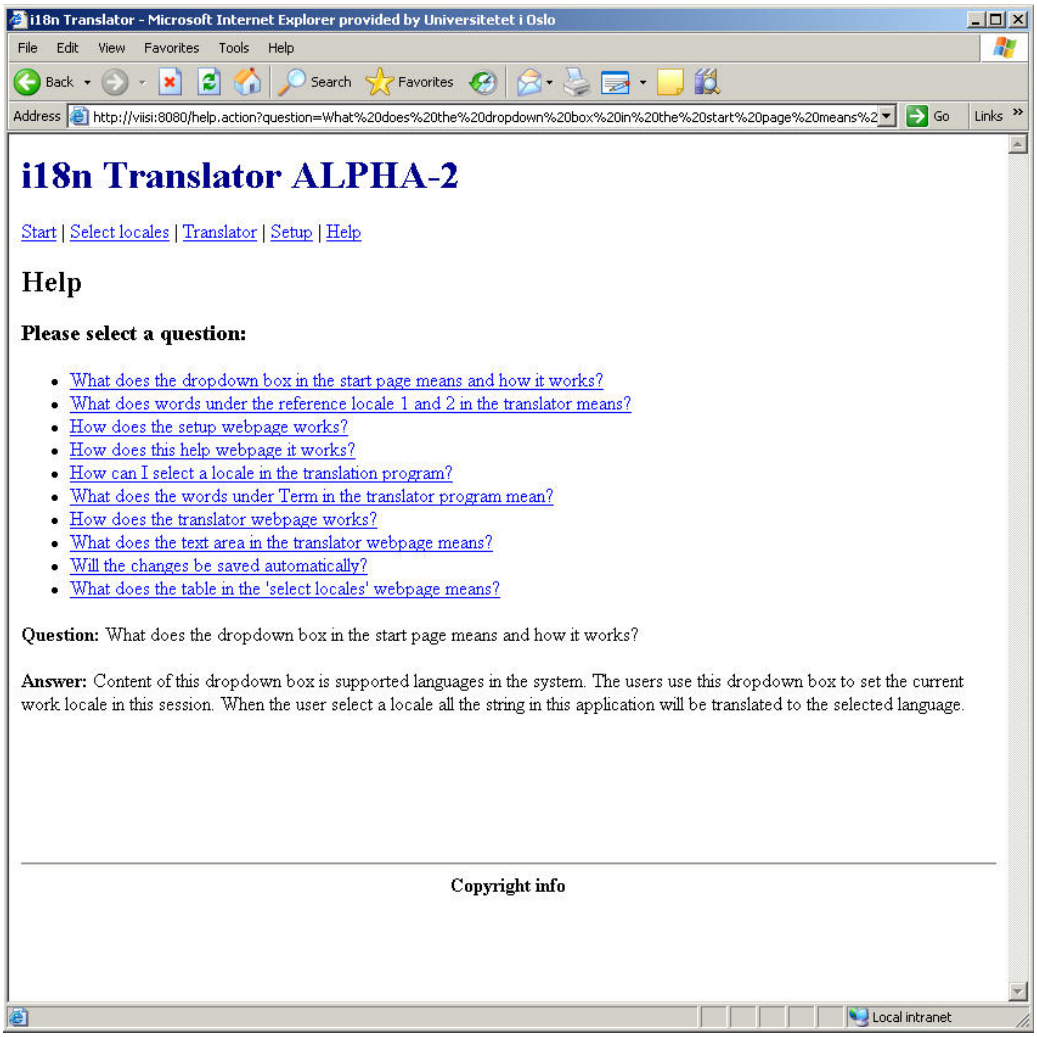

Figure 17: Help page

To help the user with the DHIS-INT module web work application, we need a little help system to answers all the questions related in the used of the application.

Øyvind create a getQuestions method to set questions (key) and corresponding answer (value) to the hash table.

```
public Hashtable getQuestions() {
  //  Set questions and corresponding answers to the hash table
  Hashtable hash = new Hashtable();
  hash.put("What does the words under Term in " +
           "the translator program mean?",
           "This is the values the system uses to identify " +
           "the translated text, they are also available for " +
           "the reference locale.");
  ...
  // Add more questions and answer to the hash table
  return hash;
}
```

Code 12: Get question method in helpAction.java class

In the velocity file, he lists all the questions from the class and link them to the correspondence answer. When the user clicks to the link on one of the questions, the requested question and value will display on the bottom of the webpage.

```
<h2>Help</h2>
<h3>Please select a question:</h3>
<ul>
#foreach ($key in $questions.keySet())
  <li><a href="help.action?question=$key">$key</a><br /></li>
#end
</ul>
#if ($question != "")
  <p><b>Question: </b>$question</p>
  <p><b>Answer: </b>$answer</p>
#end
```

Code 13: help.vm velocity file

Since I am responsible for the transaction program, it is suitable task for me to add the questions and answer related to the program. I add 10 different questions with correspondence values related to the use of this translator program.

6.8.7 Conversion of new lines and html tags and characters in the webpage

It is easy to convert new lines in html. I used only a wile-statement and loop to all the values in the hash table and convert the character '\n' to '<br>'.

```
Enumeration keys = hash.keys();
while (keys.hasMoreElements()) {
  String key = (String) keys.nextElement();
  String newValue  = (String) hash.get(key);

  while (newValue.indexOf("\n") >= 0) {
    // Getting key and value
    String value = newValue.substring(0,newValue.indexOf("\n"));
    value = value + "<br>";
    value = value + newValue.substring(newValue.indexOf("\n")+1,
newValue.length());
    newValue = value;
  }
  hash.put(key, newValue);
}
```

Code 14: Convert new lines sHTML method in GlobalService class

Even I have solved the converting all new lines for the html output, from the help webpage describe in the previous section (ref. 6.8.6) it seems that the page gets some error when I add this question to the system: *"What does words under the reference locale 1 & 2 in the translator means?"* The reason for the error was the system did not accept the '&' character in the html output, since this is one of the tags in the html. Since the time is short and it is closed to the delivery of the i18n module application, our i18n team has not enough time to implement this function on the system. It had to be one of the tasks in the future development.

6.8.8 File control class

File control is the class using to handle reading of all keys with corresponding values, which are sentences or words from a specific local property file (resource bundle), and transfer it to a hash table in the DHIS system, and to write these value from the system to the property file. Follow to this class is get existing locales methods, which get all supported locales property files in the system. We got some problems when creating this file control class.

Trouble with standard resource bundle

The first problem is during the multi-line support. Even thought Java support internationalisation by resource bundle class (ref. Resource bundle), The normal resource bundle class in java.util package doesn't have multi-line supported because the class read randomly in order for each line in the file, that's why the first line to the file has no connection with the next line, when the system read the file to the register.

Core Java and String Tokenzier

From Core Java, in the data file example they write three employees with name, salary and hire day to the file in each line, then read these data and use StringTokenzier class with '|' character to break up name, salary and hire day into each correspondence class. StringTokenzier is an existing class defined in the Java library, which used to break up sentence into separate lines. Then use the print method to display it to the console.

I used the write and read methods in Core Java to create the file control class. But it seems that we can not use the StringTokenzier class because the system defined the line terminator "\n" also as a string tokenize.

Multi-line support

Since the sentences in the resource bundle file should be multi-line support, we need a method that handles this problem. I begin first to put content of the resource bundle as property file to a specific locale into a self-defined message bundle class with keys and corresponding values as attribute and it works. Later on, one of the HISP i18n team members, Øyvind told me to return contents of these resource bundles as hash table instead. To solve this task I need to explore the hash table class (ref. 6.4.3). In the file control class the read method will read line for line from top to bottom of the file and put it into a hash table. To solve the multi-line support, Øyvind use substring method in following in the String data type and split the key and value with '=' character. Then he add this string "$###$" as line terminator in the end of each value, then use a while-loop to add lines until the final line marked with the line terminator at the end.

The following code shows how the system read the file with content of key and correspondence multiple-line value and transfers it to the system:

```
File fileName = new File("DHIS2MessageBundle_+locale+".properties");
BufferedReader in = new BufferedReader(new FileReader(fileName));

// Reading all tokens from the file
String instr = in.readLine();

// Getting key
String key = instr.substring(0,instr.indexOf("=")-1);

// Removing the key from the string
instr = instr.substring(instr.indexOf("=")+2,instr.length());
String value = "";

// Multi-line support, adding lines until the final line
// marked with line terminator at the end
while (!instr.endsWith(lineTerminator)) {
  value += instr+"\n";
  instr = in.readLine();
}
```

Code 2: Multi-line support solution

Get existing locales

Many methods and function in the DHIS-2 software need to know which locale property file support in the system. To solve this problem, we need a get existing locales method to get all existing locales in the system. First of all I create the getExisitingLocale() method based on a local. The method will return the local if the property file to the given local exists.

```
private Locale getExistingLocale(Locale locale){
/*
* Returns the current local if the proper file exists
*/
  String fileName = "DHIS2MessageBundle_"+locale+".properties";
  File file = new File(fileName);
  if (!file.exists())
    return null;
  else
    return locale;
}
```

Code 3: Get existing locale

Then I create the getExistingLocales() method. I get all available locales in the standard locale class, loop throw all the available locales and used the getExistingLocale() to find which property file exist in the system. The method returns all existent property files from the system as a locale array.

```
public Locale[] getExistingLocales()
/*
* Returns available locales if the proper file to each locale exist
*/
Locale[] existsLocales = null;

Locale[] locales = Locale.getAvailableLocales();

// Put existing locales to a dynamic array
Vector vector = new Vector();
for(int i=0; i< locales.length; i++){
  Locale l = getExistingLocale(locales[i]);
  if(l != null){
    vector.add(l);
  }
}

// Transfer the dynamic array back to a static array
existsLocales = new Locale[vector.size()];
for(int i=0; i<vector.size(); i++){
  existsLocales[i] = (Locale)vector.get(i);
}
return existsLocales;
}
```

Code 4: Get all exiting locales

6.8.9 Hibernate implementation for persistent

Since the time to this development is too short, we still have unsolved problem in DHIS-INT module, one of them is hibernate implementation for persistent.

If we have two users who visit the setup page at the same time, user A want to add new keys and value, or delete and existing key to the resource bundle in the setup webpage (ref. 6.8.5) and user B want to create new property files for an non existing local in the translator program (ref. 6.8.3), we got a big problem in persistent of data.

User A open the setup webpage, are no ready to add/delete the key in default resource bundle. At the same time, user B selects a request locale in the translate program and translate all the existing word and sentences (values) to the new language. In this time if user A add new key to the system, the default locale resource bundle (English) will now consist this new key. Content in the old keys in the translator webpage is unchanged, when user click on the translate button to create new property file to the new locale, you will miss the new key that user A just have created from default locale. The same happen if user A delete an existing key from default resource bundle.

One of the HISP i18n team is responsible in implement hibernate into the software. The problem is difficult and he still can not finish the task. I tried also to implement hibernate by create a populate database class and try to convert the flat file control store in property files

over to database with table and relationship. Since I am responsible for other unsolved tasks with higher priority, I need to concentrate to solve these first.

Some other modules have solve the implementation of hibernate into their module with successfully result. Since the result is positive, our HISP team in i18n module should also be able to solve the task if we have enough time in the future development process.

6.9 Interesting findings in a global open source project

6.9.1 OSS technologies and frameworks

The open source course at the University of Oslo is going on the first time in this semester and there are very many new technologies the participations have to learn. Much software, technologies and frameworks is new for me. Java and xml is two of the language I could in my knowledge before I started on this project. I haven't takes any open source course before, and since the project task is to develop the modules for the DHIS-2 application that no other have done it before, that's why the project is very interesting. I learn a lot of technologies and frameworks by taken this project. Spring, Hibernate, Maven, Subversion, JUnit Test and Web Work are software's I learn in the first time in this project.

Our module develops a GUI web application implement with WebWork-based software developments. Creates DAO interface for persistence, and implements it to the file control class and hash table hibernate class. We use the Xml file format to create your beans. Spring is used to map the beans from the mapping file beans.xml to this DAO interface in the system, it connect the beans file to the Java file and transfer it to the webpage. Service class is implemented to the project. This class has two scopes, global scope working in the action class and was connected to the file control class. Local scope is working in the key management with hibernate to ensure the persistence. We use maven to build the project. This software is available for running the jetty container that the web application can run online by velocity files as dynamic webpage's. JUnit Test is a simple unit test framework. It is used to test the service classes and hibernate. Eclipse is a complete IDE for software development, which is a good software development for web application. All modules in the DHIS-2 application are taking this software in use and the whole DHIS-INT module was developed by this software development. Velocity is a dynamic script language using in the web. Maven is software where we can connect the Java file together with the beans file and build a web application. This software can also run the Jetty container which generates the velocity file over to a readable website.

6.9.2 Unicode support

To enable multi-language in DHIS-2 application we need to have the Unicode support (ref. 3.1.1). In order to get the international character formatting accurate you should use Java 1.5. It gets successfully result when I save a Vietnamese sentence with Unicode symbol in the i18n WebWork-based application, the system will automatically convert the Unicode symbol to the encoded Unicode and save it in the resource bundle as '&#<xxxx>;' where <xxxx> will be characters of decimal number in the 16 bits encoded system (ref. Appendix B, 10.2), since computers used 16 bits to store a Unicode symbol. But when I saw in Microsoft Word there is some different between the codes with the same Unicode symbol, because the code in this software program is '\u<yyyy>' where <yyyy> is a combination of number and letter. I don't know why but there must be some different between these two codes. After I get an email

from Soung Tran, developer in the Report Designer module which describe she could not us the Unicode character in the Vietnamese resource bundle in the i18n module project to display it in the java framework application with swing. I search on the Internet to find out the problem. I visit the Vovisoft (ref. 3.2.5) and find the Vietnamese encoding table with Unicode (ref. Appendix B, 10.2). The result of this searching is there is several type of code to display a Unicode symbol. In the webpage the computer use the encoded Unicode decimal, but in application like Microsoft word or application in java with swing framework they use encoded Unicode hexadecimal to display the Unicode symbol. I continue with the framework application I created in the start of the project period with Vietnamese resource bundle (ref. 6.4.1) and get successfully result. Then I send her an email, with explaining of the problem and attach this application so she can test and see how the solved problem works in her computer.

6.9.3 Participation in a global software development project

Since our i18n module have communicated with the modules that are dependent of us. We need to get a team-work with different modules. After finish with the setup and development in their project, developers in other module need to internationalise their application. Several developers are asking for help of how our module work. Standard Data Entry module and Report Designer module is two of the group asked us of how the i18n works. I gave them a short demonstration of how i18n module works and explain how the system find the value in the resource bundle based on a specific key and display it to the webpage.

I get a short collaboration with Soung Tran, developer in the Report Designer module. She stays in Vietnam so we have to communicate by email. The time zone in Vietnam goes 6 hour before Norway time zone, so developer in Vietnam have almost finish with their working in the day, when we in Norway have just stay up at the bed in the morning. The collaboration work nice, she wrote questions with unsolved problems and asking me for help. I explained the problem and replay the same day. She checked her email-box in the next day and tried to solve the problem.

It is both positive and negative by working with participation in a global. If the working place is in the same place, we can talk to each other face to face. Working in a global software development project has also some positive. The result project can be use in different place, in different country with different culture. The requirements they need in Vietnam, can be in Norway they didn't need. I. E Vietnam needs Unicode-support since the Vietnamese language uses different special symbols to display a letter.

In the last period of the project I get an email from Vinh Chien Le, developer in the WP module with the follow content:

> *Dear Int group, We are using your module in our module (Ward Patient Module). It works nice except one thing, that is if we have blank line or characters after # character (lineTerminator) in resource bundle file, the system will raise an error like this: "HTTP ERROR: 500 String+index+out+of+range%3A+%2D2". So could you make some little changes to make it better?*

He gets this error because he add new key directly in the resource bundle. He should use our DHIS-INT Web Work application to add/update keys in the resource bundle, for the first our application has multi-line support and for the second it supports the Unicode symbol. Since he

stay in Vietnam I explain by writing in the mailing-list why he get that error, and I recommend he to use our application to handle the key management.

In the exploration on DHIS-1 software I get contact with Kim Anh in Vietnam by mail and yahoo messenger, which has a lot of experience in the DHIS-1 software. I still her questions with problems related to the field and got feedback from her.

6.9.4 Use of collaborative tools in software development

Subversion

Since the project is a part of the open source course, develop in an open source global project and the modules are dependent of each other, we need to share our project code with other module. We use the subversion which is a source control tool to upload the code to the repository. One way of using subversion on Windows is to run a unix-like shell, e.g installing Cygwin or x-term. Another way is to install TortoiseSVN, which integrates Subversion into Explore. We in the HISP i18n team have a little problem to upload the project code the repository in the beginning of the project period. The reason for our problem with the subversion repository was that we didn't have proper access; the responsible of the group assignment in the open source course at University of Oslo fixed this for us. In my assignment-2 delivery I use x-term to upload my code on computer at the MMCL lab. But with the i18n module project I install TortoiseSVN software on my laptop. That will be easy for me to job with the project at home.

Confluence Wiki

Since the HISPi18n module is a part of an open source project, and the users in the HISP projects and developers from the other module should understand on what our group is working with, I need to create some documentation of the work on the Confluence Wiki. This is a good collaboration tool in software development to show other group what we had done on the project. And a positive with this tool is we can exchange our knowledge and experience with participation in the HISP projects.

JIRA issue tracking system

This is another collaboration tool to create issue to the projects. The modules create necessary issue in the JIRA issue tracking system that need to be solved in the module. This is a good tool to display to all the participation in the HISP project that what each developer in every module had done, what issues is open (need to be solve), issue that have been resolve or closed.

7 Discussion

In this chapter I will address my research objective by drawing on relevant empirical findings (chapters 5 and 6) and reviewed literature (chapter 2).

As put forward in chapter 1, my research objective is:

> *To explore the importance of internationalisation in the context of information systems in developing countries, and identify challenges related to developing internationalised software in this context.*

I address this objective by extending the research area to more global context and compare my Vietnam findings with existing case studies from other internationalisation in the context of information system in developing countries where the HISP approach has been implemented. Based on the Vietnam findings and results from the global comparisons, I discuss the importance of internationalisation and the challenges with internationalisation in the HISP implementation in developing countries in general and Vietnam in particular.

The following sections will be presented:

7.1 – The importance of internationalisation in developing countries

7.2 – Challenges with internationalisation

7.1 The importance of internationalisation in developing countries

There have been many efforts to apply IT in developing countries and unfortunately a large part of the projects have failed.

Lack of focus on local adaptation and local language

A common problem seems to be the pure technological focus with emphasis on technical features and technological hypes. The lack of focus on the local context where the information system is implemented has been pointed out as an important factor for failure.

Reference to IT in developing country, adaptation is an important process in the information technology transfer life cycle (ref. 2.1). The adaptations covered three areas: performance enhancement, new functions, and match to local conditions. Most of the adaptations could be characterised as adding functions to organisational context, which often required quite specialised knowledge. In other cases, the adaptations involved software development as a part of information system. The strategy for the design of the DHIS in South Africa was based on a set of objectives and scenarios presented by Kaasbøll and Nhamossa (2000). The importance lack among these objectives, scenarios and HISP implementation are adaptation to match local condition and support the local language.

Let us take a look at the Vietnam context. Vietnamese is the official language in Vietnam, being used for communication in both public and private sector. The DHIS was developed using English standards, implying that the application and its documentation are written in English. In order to start using the system in the Vietnamese context, the system should be adjusted to fit the health information processes and the translation from English to Vietnamese is required. The adaptation of IS in context from one country to another country

faces many challenges and problems like the limitation in technically platform, domain knowledge, different social cultures and challenges in translation of different languages.

Both Kaasbøll and Nhampossa (2000), and Bark and Heeks (1999) propose to increase focus on assimilation and local adaptation of the transferred technology (ref. section 2.1). Support for the local language is a key here so that internationalisation of the transferred software is needed to be able to adapt to the new context. So we see that internationalisation is of great importance to enable local adaptation.

Networks of action demands support for many different contexts

As put forward in chapter 3, the HISP goal is:

> *"The primary goal of the HISP research is to design, implement, and sustain HIS following a participatory approach to support local management of health care delivery and information flows in selected health facilities, districts, and provinces, and its further spread within and across developing counties."* (Braa et al. 2004)

The networks of action approach (Braa et al. 2004) addresses the problem of sustainability and scalability, where Braa et al. (2004) describe how experiences, knowledge, technology, and value create opportunities for sharing between the various nodes by establishing networks.

The point with the establishing the network of users in several development countries and by the networks developers builds up the supporting which doing the system more sustainable. The network of countries like the nodes in the HISP network is a strategy to get the IS more successful and sustainable, by sharing experiences and knowledge, including software and IT implementation experiences. To get success result in the field, we need internationalisation of global software.

By developing local software tailored to one specific context the software can only be used in exactly the requested location. Such a local approach limits the possibilities to establish networks of action across different contexts and thereby limiting the chances for sustainability. In the HISP context such an approach would mean that the developers would have to start development of software from scratch in each new national context.

By going for a global software approach where the same software can used in different countries enables the transferring of this software and implementation strategies to different countries and supports the creation of networks of action. The HISP implementations in different development countries follow such an approach and the DHIS software provides the possibility to adapt to each new local context.

Such an approach is also supported by how Nielsen and Nhanpossa (2005) define internationalisation; as a process with the aim to reduce the costs of development support the reuse of technical and human resource and building, and sharing of knowledge across national borders, and more technically, the DHIS software aligns with Barbour's understanding of internationalisation as the process of making application software independent or transparent to natural language (Barbour 1996).

7.2 Challenges with internationalisation of software

Design for internationalisation

HISP implementation was started in South Africa, and the first software development processes of HISP project had no intention of developing global software, the DHIS was supposed to be used only in the South African context, and tailored to the Western Cape province. After the HISP's initial success in the Western Cape and later nation-wide in South Africa, the HISP project has been adopted in other countries. Mozambique was the first country to adopt the South African project. At the beginning of the Mozambican DHIS implementation all user interface forms and objects were hard-coded. Translating the software from English to Portuguese was required by the Mozambican Ministry of Health, and this process faced many problems. The first version with hard-coded language translation was prototyped in northern Mozambique in 2000.

In the technical view, it is not easy to enable multi-language support in software applications. The transfer of DHIS from South Africa to Mozambique proved very difficult and especially the translation process as the user interface of the DHIS was based on hard text strings. This approach of hard-coded text strings contradicts Abramson's advice to internationalisation of software:

> *"The best design approach to internationalise software is to maintain strict language independence throughout the code."* (Abramson, 1994)

To the HISP developers' defence it should be said that the DHIS was developed with the objective to support the local health services in Western Cape province in South Africa and they had never dreamt of an international use of the DHIS.

The first i18n process in Mozambique lead to a reconstruction of the DHIS architecture with an own language database that decouples the text strings from the code.

The initiation of the HISP implementation in Mozambique is an example of how difficult it is to internationalise the software when i18n support is not part of the design. But the HISP team learned a lot from this problematic experience in Mozambique and with the i18n support as part of the architecture most of the later DHIS adaptation processes has been far more successful with regards to local language support.

Domain knowledge important

To translate the software from a language to another has a lot of challenges and difficulties. It influences the culture, technical knowledge, doctors and developers. If a technical developer translates the DHIS software, there are many words and expressions in the health sector that the developer most likely does not understand. On the other side, if a doctor translates the DHIS software, he would probably not understand some of the more technical issues related to the software. Based on the implementation experiences from the HISP project it seems like a cooperation between the doctor or a domain expert in general and the developer is a good solution.

The lack of knowledge from developers influences development of the software. A large part of methods and action in the system needs spread experience and lot of knowledge to finish the job.

The culture and the locale have also many challenges. Many health services have different organisation structure, with different levels from national, province, districts to the primary services. For example the hierarchical organizational structure of the health system in South Africa has five levels; in comparison with Mozambique has one less level. (Nhampossa, 2004)

Limitations in the technological platform

From the HISP-Vietnam case we have seen that even though i18n support is part of the DHIS architecture the limitations in the technical platform have made the Vietnamese translation very complicated (ref 5.3). Not all operative systems are fully internationalised, for example from the Windows platform, only Windows XP has Vietnamese in the regional setting. As described earlier, using DHIS with earlier Windows versions, the system needs to store the Vietnamese text strings under another location (English) as a work-around. By taking this solution in use, the software will miss the intent to internationalisation, the system do not specify to the correct locale and the user need to have some experience with the software to take this software in use. Another problem is related to Ms-Access, Ms-Excel and Visual Basic 6.0. Even though the Ms-Office tools can write, display and store Unicode characters; the Visual Basic does not directly support Unicode. Although VB 6.0 stores string internally as Unicode it has several limitations (ref. 3.2.5). Properties Window in IDE, property bag, clipboard function and menus are ANSI only, and the converted of Unicode strings are displayed as '????'.

All Ms-Software is of course proprietary and hence the source code is not open, such as Windows, Ms-Office tools and VB 6.0. Open Office is an Open Source alternative to MS Office. The Open Office source code is open and available to any interested developer and anyone can join the community and start to translate software or documentation. Open Office have more supported language than Microsoft Office, and the Vietnamese language is one of the supported language in Open Office.

The VI.OpenOffice.org project in the Vietnamese language is a part of the Open Office, which have the goal to contribute the project and the software to the community which use the Vietnamese language. The VI.OpenOffice.org project has some related contents with the first open source software (Open Office, Vietnamese):

- Develop and design the home page
- Collect all translate Vietnamese document and resource
- Search all other homepage which have this software download
- Strengthen the Vietnamese translation
- Develop and strengthen the language tools, for example correct the word spelling
- Diffuse the open office in the Vietnamese community

The DHIS-2 version is based on the Ms-Office tools DHIS-1 software. The DHIS-2 version is still under development, and it is based on the Java Platform which is a Free and Open Source Software (FOSS). Almost all of the problems in development of DHIS-1 software will disappear in the new version DHIS-2. Even many problems in DHIS-1 is solved in DHIS-2 software, it will of course appear some other problems, display special tags in the web browser and converting from Unicode decimal to Unicode hexadecimal character, etc. are problems that are yet to be solved.

Limited support for Vietnamese language

After all the struggling with the Vietnamese translation of the DHIS-1 it is interesting to ask why the Vietnamese language is not properly supported by Microsoft? Many languages, countries neighbouring to Vietnam are supported in the Microsoft Office tools, for example Chinese, Laos, Thailand and Cambodia (ref. 3.2.5). The problems in use of Unicode in the Microsoft Office tool are related to the Vietnamese language. Vietnam has had a long history of oppression. The country has been a province of China for over 1000 years, from 1859 Vietnam was more or less colonised by the French and around 20 years with war between the Communists in the north and the Liberates in the south from 1954 (Florence and Jealouse 2003). Another opinion is described earlier from Vovisoft (ibid), one of the Vovisoft colleges has this mention in the end of 1999:

> *"Don't bother with Unicode yet. There's no such thing yet within Microsoft. Probably, they have abandoned Vietnam because they could not make any software sales in Vietnam."* (Vovisoft a)

Maybe oppression and the point made by the Vovisoft above have influenced the political reason. Another reason might be the historical relationship between Vietnam and the United States. The US trade embargo on Vietnam up until 1994 and of course the war between north and south, where the south was heavily supported by the United States and allies in the Free World Military Forces. With the political reason and relation with the United States, the limited support for the Vietnamese characters is still the problem in the Vietnamese language.

From the technical side, the results from the Vovisoft team search (ref. 3.2.5), the font like Arial, Tahoma and Verdana have glyphs of Vietnamese characters, but the font size must be over 170 kb or have a time stamp after June 1998. As we can see, we can not find any glyphs of the Vietnamese characters before 1998 and development in use of Unicode in the Vietnamese letter is started not before the end of 1999. Based on this discussion the problems are related both to political, business and technical reasons.

Problems from other IT-based projects

There are several other IT projects with some problems of supported language use in information systems. Among these are examples from the Devenagari language in Nepal and the Amharic language in Ethiopia.

From the final report (Nepal Health, 2005) in Nepal Health Monitoring Pilot Project with the Devenagari language it says:

> *"It was possible to present the forms in the local language and script, although the programming required significantly more time from a higher level of skilled computer programs than anticipated and makes it difficult at this point to have this process transferred to the non-profits without support."* (Nepal Health, 2005)

The Amharic language in Ethiopia has a complex script that is not supported by MS Windows or MS Office. The DHIS software has never been translated to Amharic in Ethiopia because of the lack of the Amharic language support in Microsoft, but luckily almost all the users in Ethiopia are fluent in English so there is no big problem related to the language yet. Interestingly you can find support for Amharic in Open Office 2.0.

From these examples you can see Microsoft has been prioritizing certain languages in favour of others. The reasons for why Microsoft does this might be explained by looking at the business in the market, on which countries that invest most in the Microsoft products.

It seems like there is a more extensive support for local languages and scripts in the western world than in developing countries, not surprising as there are generally huge differences in IT industry and local IT usage between developed and developing countries. However, as experienced in the HISP Vietnam case, these differences makes it a lot more complicated to provide locally adapted software in developing countries.

8 Conclusion

In this chapter I will summarise my research and make some concluding remarks on the most important aspects of multi-language software support in developing countries in relation to my research objective:

> *To explore the importance of internationalisation in the context of information systems in developing countries, and identify challenges related to developing internationalised software in this context.*

Implementation of computerised information systems in developing countries has proven difficult (Krishna and Walsham 2005, Bark and Heeks 1999). One important reason has been too much focus on technology and a lack of adaptation in the local context. Establishing networks of action is a suggestion strategy to improve the situation. Global networks of action like the HISP project have a need for internationalised global software, but it seems like this is often difficult to develop.

To design for internationalisation is an important first step. It is not easy to internationalise the software when it is not a part in the software design, and the history of internationalisation in the HISP project is a good example of how the local adaptation process can be improved when the 18n support is a part of the architecture.

Furthermore, choice of **technological platform** has proven decisive when it comes to i18n support. When the local language is not supported in the proprietary software such as MS Windows, Ms-Office tools and VB 6.0 you have no choice but to wait. Open source software can be a solution to this problem, and software such as Open Office or Open Source Java technologies seems to provide a more extensive support for languages and more important, they have an open community-based internationalisation process.

There seems to be **poorer language support in development countries**, in this thesis exemplified by the Vietnamese language. Other examples have been the Nepali language Devenagari and the Ethiopian language Amharic. This problem is related to the choice of technological platform as describe above as open source alternatives has proven more including most likely due to the open translation process. The limited support for the Vietnamese language in Microsoft products, especially in the older versions, might be influenced by politics, with the oppression in the Vietnamese history and the business situation.

Even though HISP had not yet solved the Vietnamese language problem in the DHIS-1, the Vovisoft team has created a program to *convert from encoding font to Unicode characters* in text-file format with successfully result. I think this solution might be able to solve the DHIS problem as well if the HISP project can accept a simplified version of DHIS-1 software.

I turn to my own research and the difficulties experienced, and conclude that my **action research** approach created a lot of knowledge I could not have achieved in any other way. Only through close collaboration and action with local stakeholders, pursuing a solution to a real problem, could I acquire the local knowledge and insights necessary to understand the problem area. While I sometimes was restricted in movement and collaboration due to my base in Norway, my participation in the HISP project gave me an understanding of the problem that would not have been detectable from “the laboratory”.

The HISP network has produced a range of action research case studies from several development countries which have been valuable to my research and understanding of the theory and internationalisation support in information systems development in developing countries. This master study has provided many interesting findings, and through comparison with other studies I have been able to generalise some ideas and thus I hope, also contributed to the field of internationalisation in information system.

9 References

Abrahamson, 1994:
Abrahamson D.
Globalization of Windows, BYTE, Vol 19. No 6. 177-184

Bonté
La Bonté, A.
Personal communication. Secrétariat du Conseil du treésor du Québec, Canada

Braa and Hedberg, 2002
The Struggle for District-Based Health Information Systems in South Africa

Braa et al. 2004
Jørn Braa, Eric Monteiro, Sundeep Sahay
"Networks of Action: Sustainable Health Information System Across Developing Countries"

Braa, undated
Strategies for developing health information system in developing countries

Core Java
Example 12-2, Volume I, Fundamentals, The Sun Microsystems Press Java Series, Cay S. Horstmann, Gary Cornell, ISBN 0-13-081933-6

DHIS-1.3, 2005
[Visit 14.10.2005]
http://www.hisp.info/confluence/display/DHIS1/DHIS+1.3

JAARS Museum, 2003:
JAARS Museum of the alphabet: Vietnamese Alphabet
[Visit 06.10.2005]
http://www.jaars.org/museum/alphabet/galleries/vietnamese.htm

Krishna and Walsham, 2005
Implementing Public Information Systems in Developing Countries:
"Learning From a Success Story"

Kaasbøll and Nhampossa, 2002
Transfer of public sector information systems between developing countries: south-south cooperation. Proceeding of IFIP 9.4. "Social Implication of Computers in Developing Countries"

Nepal Health, 2005
Nepal Health Monitoring Pilot Project, Final Report
http://www.healthnet.org

Nhampossa, 2004
Nhampossa, José Leopoldo
The challenge of "Translating" health information systems from one developing country context to another: case study from Mozambique

Magnset, 2005
Lars Mangset Master thesis
"DHIS-2 A Globally Distributed Development Process"

Open Office
http://www.openoffice.org/index.html

Open Office, Vietnamese
http://vi.openoffice.org/

Java Internationalisation
Sun Development Network
http://java.sun.com/j2se/corejava/intl/index.jsp

Java Tutorial A:
Backing a ResourceBundle with Properties Files, Sun Development Network
Available at:
http://java.sun.com/docs/books/tutorial/i18n/resbundle/propfile.html

Java Tutorial B:
Identifying Available Locales, Sun Development Network
Available at:
http://java.sun.com/docs/books/tutorial/i18n/locale/identify.html

Tran et al. 1997
Tran Luu Chuong, Nguyen Hoang, Ngo Thanh Nhan, Do Ba Phuoc, Ngo Trung Viet
Country Report – Vietnam,
"Current status of Vietnamese language processing and multilingual processing"
[Visit 18.10.2005] Available at
http://www.informatik.uni-leipzig.de/~duc/software/misc/viet.html

Vietnamese Online, 1997
[Visit 10.10.2005] Available at:
http://www.geocities.com/CollegePark/Campus/6336/index.html#LessonIndex

Vovisoft a
Unicode history
[Visit 28.09.2005] Available at:
http://www.vovisoft.com/unicode/1yearunicode.htm

Vovisoft b
Unicode converter program
[Visit 28.09.2005] Available at:
http://www.vovisoft.com/VisualBasic/UnicodeConvert.htm

Vovisoft c
Vietnamese JavaScript Input Editor
http://www.vovisoft.com/unicode/utilities/vietjie/default.asp

Vovisoft d
Vietnamese encoding table with Unicode
[Visit 28.09.2005] Available at:
http://www.vovisoft.com/unicode/utilities/VietEncoding.htm

VIQR, 1996
[Visit 17.10.2005] Available at:
http://vietunicode.sourceforge.net/charset/viqr.html

Unicode
[Visit 28.09.2005] Available at:
http://www.unicode.org/standard/WhatIsUnicode.html

10 Appendix

10.1 Appendix A: Files and folders structure

```
dhis-i18n
+-- src
|   +-- main
|   |   +-- java
|   |   |   +-- org.hisp.dhis.i18n                  // module class
|   |   |   |   +-- I18n.java
|   |   |   |
|   |   |   +-- org.hisp.dhis.i18n.action           // action classes
|   |   |   |    +-- HelpAction.java
|   |   |   |    +-- LocaleAction.java
|   |   |   |    +-- MessageBundle.java
|   |   |   |    +-- SaveActon.java
|   |   |   |    +-- SetLocaleAction.java
|   |   |   |    +-- SetupAction.java
|   |   |   |    +-- StartAction.java
|   |   |   |    +-- TranslateAction.java
|   |   |   |    +-- TranslatorAction.java
|   |   |   |
|   |   |   +-- org.hisp.dhis.i18n.dao              // dao interface
|   |   |   |     +-- HashtableDao.java
|   |   |   |
|   |   |   +-- org.hisp.dhis.i18n.dao.flat         // flat implementation
|   |   |   |     +-- FileControl.java
|   |   |   |
|   |   |   +-- org.hisp.dhis.i18n.dao.hibernate    // hibernate implementation
|   |   |   |     +-- HibernateHashtableDao.java
|   |   |   |
|   |   |   +-- org.hisp.dhis.i18n.dao.model
|   |   |   |     +-- DhisResourceBundle.java
|   |   |   |
|   |   |   +-- org.hisp.dhis.i18n.dao.service
|   |   |        +-- Service.java          // Abstract class that services inherit from
|   |   |        +-- LocalService.java     // Local service implementation
|   |   |        +-- GlobalService.java    // Global service implementation
|   |   |
|   |   +-- resources
|   |       +-- ResourceBundles
|   |       |    +-- local                // directory that contain local resources
|   |       |    +-- global               // directory that contain global resources
|   |       +-- META-INF
|   |       |     +-- dhis
|   |       |          +-- beans.xml      // Spring bean configuration
|   |       |
|   |       +-- validators.xml
|   |       +-- xwork.xml                 // action mapping
|   |
|   +-- webapp
|       +-- WEB-INF
|       |   +-- web.xml                   // dispatcher description
|       |
|       +-- bottom.vm                     // velocity
|       +-- help.vm
|       +-- index.html
|       +-- index.vm
|       +-- save.vm
|       +-- select_locale.vm
|       +-- setup.vm
|       +-- start.vm
|       +-- style.css
|       +-- top.vm
|       +-- translate.vm
|       +-- transaltor.vm
|
+-- build.properties
+-- jetty-config.xml
+-- plugin.xml
+-- project.xml
+-- velocity.log
```

Module class

- I18n.java: The i18n module class.

Action class

- HelpAction.java: Used by velocity file "help.vm" to provides a little help system to the translator through the xwork mapping "help.action".

- LocaleAction.java: Used by velocity file "select_locale.vm" to get a list of all available locales through the xwork mapping "getlocales.action".

- MessageBundleAction.java: Used by velocity file "setup.vm" to get the selected action from the users through the xwork mapping "messagebundle.action". Based on the selected action the system will add, update or delete a key in the default property file.

- SaveActon.java: Used by velocity file "translator.vm" to save keys and translated words/sentences values entered by users through the xwork mapping "save.action". Recieves CGI parameters wrapped in a map and must therefore implement ParameterAware. The alternative method using setters and getters would not work because the CGI parameters are arrays which can be of different lengths. We loop through the arrays and collect associated CGI parameters and put the keys and values in to the locale hashtable. If the property file to the current locale is available on the system we update the keys with corresponding values, if not we creates a new file and save keys with corresponding values to the file.

- SetLocaleAction.java: Used by velocity file "select_locale.vm" to set the locale, reference locale 1 & 2 to the session through the xwork mapping "setlocale.action".

- SetupAction.java: Used by velocity file "setup.vm" to provides a little setup system to the default property file through the xwork mapping "setup.action".

- StartAction.java: Used by velocity file "start.vm" to set the default and selected work locales in to the session through the xwork mapping "start.action".

- TranslateAction.java: Use by velocity file "translator.vm" to get content of the current properties file the users just have saved or updated througth the xwork mapping "translator.action" from the xwork action "save.action".

- TranslatorAction.java: Use by the velocity file "translator.vm" to get content of the selected locale, reference locale 1 & 2 from the properties files and return these as a hashtable through the xwork mapping "translator.action".

Flat implementation

- FileControl.java: Use by the service and action package to get content of a property file as hash table based on given locale and save the keys with corresponding values from given locale to the property file. The file will also get existing locales in the system, add, update and delete a key in the default property file through the action class "setupAction.java".

Message bundle

- DHIS2MessageBundle_locale.properties: Content of keys with coresponding values to all existing locales.

Action mapping

- xwork.xml: Map the velocity files to action classes.

Dispatcher description

- web.xml

Velocity

- bottom.vm: The bottom of a page.
- help.vm: A little help system with questions and answers that help users to use the i18n module WebWork-based application.
- index.html: Start of the web application. Link to start the start.action.
- select_locale.vm: Select the locale you currently working on, and an optional reference locale 1 and 2.
- setup.vm: A little setup system where users can add, update or delete a key in the default property file.
- start.vm: Get the default work locale. You can select other locale in a dropdown list. Select of a locale, all the string on the i18n module WebWork-based application will be translated to the selected locale language.
- style.css: Style sheet to the web application.
- top.vm: The top of a page.
- translate.vm: Give feedback to the users that current properties file is saved/updated successfully and display content of this file to the webpage.
- translator.vm: This is the translator program where the users can:

  - see words/sentences to be translated (English or exists language).

  - translate of these words in to a requested language.

10.2Appendix B: Vietnamese encoding table with Unicode

VIQR	TCVN	TCVN Hex	Unicode Symbol	Unicode Hex Dec	UTF-8 Hex		VIQR	TCVN	TCVN Hex	Unicode Symbol	Unicode Hex Dec	UTF-8 Hex
a'	¸	B8	á	00E1 225	C3 A1		A'	A¸	41 B8	Á	00C1 193	C3 81
a`	µ	B5	à	00E0 224	C3 A0		A`	Aµ	41 B5	À	00C0 192	C3 80
a?	¶	B6	ả	1EA3 7843	E1 BA A3		A?	A¶	41 B6	Ả	1EA2 7842	E1 BA A2
a~	·	B7	ã	00E3 227	C3 A3		A~	A·	41 B7	Ã	00C3 195	C3 83
a.	¹	B9	ạ	1EA1 7841	E1 BA A1		A.	A¹	41 B9	Ạ	1EA0 7840	E1 BA A0
a(	¨	A8	ă	0103 259	C4 83		A(	¡	A1	Ă	0102 258	C4 82
a('	¾	BE	ắ	1EAF 7855	E1 BA AF		A('	¡¾	A1 BE	Ắ	1EAE 7854	E1 BA AE
a(`	»	BB	ằ	1EB1 7857	E1 BA B1		A(`	¡»	A1 BB	Ằ	1EB0 7856	E1 BA B0
a(?	¼	BC	ẳ	1EB3 7859	E1 BA B3		A(?	¡¼	A1 BC	Ẳ	1EB2 7858	E1 BA B2
a(~	½	BD	ẵ	1EB5 7861	E1 BA B5		A(~	¡½	A1 BD	Ẵ	1EB4 7860	E1 BA B4
a(.	Æ	C6	ặ	1EB7 7863	E1 BA B7		A(.	¡Æ	A1 C6	Ặ	1EB6 7862	E1 BA B6
a^	©	A9	â	00E2 226	C3 A2		A^	¢	A2	Â	00C2 194	C3 82
a^'	Ê	CA	ấ	1EA5 7845	E1 BA A5		A^'	¢Ê	A2 CA	Ấ	1EA4 7844	E1 BA A4
a^`	Ç	C7	ầ	1EA7 7847	E1 BA A7		A^`	¢Ç	A2 C7	Ầ	1EA6 7846	E1 BA A6
a^?	È	C8	ẩ	1EA9 7849	E1 BA A9		A^?	¢È	A2 C8	Ẩ	1EA8 7848	E1 BA A8
a^~	É	C9	ẫ	1EAB 7851	E1 BA AB		A^~	¢É	A2 C9	Ẫ	1EAA 7850	E1 BA AA
a^.	Ë	CB	ậ	1EAD 7853	E1 BA AD		A^.	¢Ë	A2 CB	Ậ	1EAC 7852	E1 BA AC
e'	Đ	D0	é	00E9 233	C3 A9		E'	EĐ	45 D0	É	00C9 201	C3 89
e`	Ì	CC	è	00E8 232	C3 A8		E`	EÌ	45 CC	È	00C8 200	C3 88
e?	Î	CE	ẻ	1EBB 7867	E1 BA BB		E?	EÎ	45 CE	Ẻ	1EBA 7866	E1 BA BA
e~	Ï	CF	ẽ	1EBD 7869	E1 BA BD		E~	EÏ	45 CF	Ẽ	1EBC 7868	E1 BA BC
e.	Ñ	D1	ẹ	1EB9 7865	E1 BA B9		E.	EÑ	45 D1	Ẹ	1EB8 7864	E1 BA B8
e^	ª	AA	ê	00EA 234	C3 AA		E^	£	A3	Ê	00CA 202	C3 8A
e^'	Õ	D5	ế	1EBF 7871	E1 BA BF		E^'	£Õ	A3 D5	Ế	1EBE 7870	E1 BA BE
e^`	Ò	D2	ề	1EC1 7873	E1 BB 81		E^`	£Ò	A3 D2	Ề	1EC0 7872	E1 BB 80
e^?	Ó	D3	ể	1EC3 7875	E1 BB 83		E^?	£Ó	A3 D3	Ể	1EC2 7874	E1 BB 82
e^~	Ô	D4	ễ	1EC5 7877	E1 BB 85		E^~	£Ô	A3 D4	Ễ	1EC4 7876	E1 BB 84
e^.	Ö	D6	ệ	1EC7 7879	E1 BB 87		E^.	£Ö	A3 D6	Ệ	1EC6 7878	E1 BB 86
i'	Ý	DD	í	00ED 237	C3 AD		I'	IÝ	49 DD	Í	00CD 205	C3 8D
i`	×	D7	ì	00EC 236	C3 AC		I`	I×	49 D7	Ì	00CC 204	C3 8C
i?	Ø	D8	ỉ	1EC9 7881	E1 BB 89		I?	IØ	49 D8	Ỉ	1EC8 7880	E1 BB 88
i~	Ü	DC	ĩ	0129 297	C4 A9		I~	IÜ	49 DC	Ĩ	0128 296	C4 A8
i.	Þ	DE	ị	1ECB 7883	E1 BB 8B		I.	IÞ	49 DE	Ị	1ECA 7882	E1 BB 8A
o'	ã	E3	ó	00F3 243	C3 B3		O'	Oã	4F E3	Ó	00D3 211	C3 93
o`	ß	DF	ò	00F2 242	C3 B2		O`	Oß	4F DF	Ò	00D2 210	C3 92
o?	á	E1	ỏ	1ECF 7887	E1 BB 8F		O?	Oá	4F E1	Ỏ	1ECE 7886	E1 BB 8E
o~	â	E2	õ	00F5 245	C3 B5		O~	Oâ	4F E2	Õ	00D5 213	C3 95
o.	ä	E4	ọ	1ECD 7885	E1 BB 8D		O.	Oä	4F E4	Ọ	1ECC 7884	E1 BB 8C
o^	«	AB	ô	00F4 244	C3 B4		O^	¤	A4	Ô	00D4 212	C3 94
o^'	è	E8	ố	1ED1 7889	E1 BB 91		O^'	¤è	A4 E8	Ố	1ED0 7888	E1 BB 90
o^`	å	E5	ồ	1ED3 7891	E1 BB 93		O^`	¤å	A4 E5	Ồ	1ED2 7890	E1 BB 92
o^?	æ	E6	ổ	1ED5 7893	E1 BB 95		O^?	¤æ	A4 E6	Ổ	1ED4 7892	E1 BB 94
o^~	ç	E7	ỗ	1ED7 7895	E1 BB 97		O^~	¤ç	A4 E7	Ỗ	1ED6 7894	E1 BB 96
o^.	é	E9	ộ	1ED9 7897	E1 BB 99		O^.	¤é	A4 E9	Ộ	1ED8 7896	E1 BB 98
o+	¬	AC	ơ	01A1 417	C6 A1		O+	¥	A5	Ơ	01A0 416	C6 A0

o+'	í	ED	ớ	1EDB 7899	E1 BB 9B		O+'	¥í	A5 ED	Ớ	1EDA 7898	E1 BB 9A
o+`	ê	EA	ờ	1EDD 7901	E1 BB 9D		O+`	¥ê	A5 EA	Ờ	1EDC 7900	E1 BB 9C
o+?	ë	EB	ở	1EDF 7903	E1 BB 9F		O+?	¥ë	A5 EB	Ở	1EDE 7902	E1 BB 9E
o+~	ì	EC	ỡ	1EE1 7905	E1 BB A1		O+~	¥ì	A5 EC	Ỡ	1EE0 7904	E1 BB A0
o+.	î	EE	ợ	1EE3 7907	E1 BB A3		O+.	¥î	A5 EE	Ợ	1EE2 7906	E1 BB A2
u'	ó	F3	ú	00FA 250	C3 BA		U'	Uó	55 F3	Ú	00DA 218	C3 9A
u`	ï	EF	ù	00F9 249	C3 B9		U`	Uï	55 EF	Ù	00D9 217	C3 99
u?	ñ	F1	ủ	1EE7 7911	E1 BB A7		U?	Uñ	55 F1	Ủ	1EE6 7910	E1 BB A6
u~	ò	F2	ũ	0169 361	C5 A9		U~	Uò	55 F2	Ũ	0168 360	C5 A8
u.	ô	F4	ụ	1EE5 7909	E1 BB A5		U.	Uô	55 F4	Ụ	1EE4 7908	E1 BB A4
u+		AD	ư	01B0 432	C6 B0		U+	¦	A6	Ư	01AF 431	C6 AF
u+'	ø	F8	ứ	1EE9 7913	E1 BB A9		U+'	¦ø	A6 F8	Ứ	1EE8 7912	E1 BB A8
u+`	õ	F5	ừ	1EEB 7915	E1 BB AB		U+`	¦õ	A6 F5	Ừ	1EEA 7914	E1 BB AA
u+?	ö	F6	ử	1EED 7917	E1 BB AD		U+?	¦ö	A6 F6	Ử	1EEC 7916	E1 BB AC
u+~	÷	F7	ữ	1EEF 7919	E1 BB AF		U+~	¦÷	A6 F7	Ữ	1EEE 7918	E1 BB AE
u+.	ù	F9	ự	1EF1 7921	E1 BB B1		U+.	¦ù	A6 F9	Ự	1EF0 7920	E1 BB B0
y'	ý	FD	ý	00FD 253	C3 BD		Y'	Yý	59 FD	Ý	00DD 221	C3 9D
y`	ú	FA	ỳ	1EF3 7923	E1 BB B3		Y`	Yú	59 FA	Ỳ	1EF2 7922	E1 BB B2
y?	û	FB	ỷ	1EF7 7927	E1 BB B7		Y?	Yû	59 FB	Ỷ	1EF6 7926	E1 BB B6
y~	ü	FC	ỹ	1EF9 7929	E1 BB B9		Y~	Yü	59 FC	Ỹ	1EF8 7928	E1 BB B8
y.	þ	FE	ỵ	1EF5 7925	E1 BB B5		Y.	Yþ	59 FE	Ỵ	1EF4 7924	E1 BB B4
dd	®	AE	đ	0111 273	C4 91		DD	§	A7	Đ	0110 272	C4 90